Soluciones de AWS

Construcciones de Soluciones de AWS

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

Construcciones de Soluciones de AWS: Soluciones de AWS

Copyright © 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 comerciales 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

Información general ... 1
¿Qué es AWS Solutions Constructs? ... 1
¿Por qué utilizar AWS Solutions Constructs? ... 1
Introducción ... 3
Requisitos previos ... 3
Instalación del CDK de AWS ... 4
Uso de componentes de soluciones de AWS ... 4
Tutorial - Parte 1 ... 5
Construcciones de Hello ... 5
Creación del directorio de aplicaciones e inicialización del CDK de AWS ... 6
Actualizar dependencias de la base del proyecto ... 7
Código del controlador Lambda ... 9
Instalar las dependencias de AWS CDK y AWS Solutions Constructs ... 10
Agregar un patrón de Amazon API Gateway/AWS Lambda a su pila ... 12
Implementar cdk ... 18
Salida de la pila ... 19
Probar su aplicación ... 19
Tutorial - Parte 2 ... 19
Código de Hit Counter Lambda ... 20
Instalar las nuevas dependencias ... 22
Defina los recursos ... 23
Revisar los cambios ... 36
cdk/ ... 37
Salida de la pila ... 38
Probar su aplicación ... 38
Casos de uso de ejemplo ... 39
Sitio web estático de AWS S3 ... 40
Controlador de imágenes sencillo sin servidor de AWS ... 40
Aplicación web sin servidor de AWS ... 40
Referencia de la API ... 42
Modules ... 42
Contenido Module ... 42
aws-apigateway-dynamodb ... 43
Información general ... 43

Inicializador ... 44
Patrón de construcción ... 44
Propiedades de patrón ... 46
Configuración predeterminada ... 47
Arquitectura ... 48
GitHub ... 48
aws-apigateway-iot ... 48
Información general ... 49
Initializador ... 50
Patrón de construcción ... 50
Propiedades de patrón ... 51
Configuración predeterminada ... 52
Arquitectura ... 55
Ejemplos ... 55
GitHub ... 57
aws-apigateway-kinesisstreams ... 57
Información general ... 58
Initializer ... 58
Patrón de construcción ... 59
Propiedades de patrón ... 60
Ejemplo de uso de las API ... 61
Configuración predeterminada ... 62
Arquitectura ... 63
GitHub ... 63
aws-apigateway-lambda ... 63
Información general ... 64
Initializador ... 64
Patrón de construcción ... 65
Propiedades de patrón ... 66
Configuración predeterminada ... 66
Arquitectura ... 67
GitHub ... 68
aws-apigateway-sagemakerendpoint ... 68
Información general ... 68
Initializer ... 69
Patrón de construcción ... 70

Propiedades de patrón 71
Uso de las API de ejemplo 61
Configuración predeterminada 72
Arquitectura 73
GitHub 73
aws-apigateway-sqs 73
Información general 74
Inicializador 74
Patrón de construcción 75
Propiedades de patrón 77
Ejemplo de uso de las API 61
Configuración predeterminada 78
Arquitectura 79
GitHub 79
aws-cloudfront-apigateway 80
Información general 80
Inicializador 81
Patrón de construcción de construcción 82
Propiedades de patrón 82
Configuración predeterminada 83
Arquitectura 84
GitHub 84
aws-cloudfront-apigateway-lambda 84
Información general 85
Inicializador 86
Patrón de construcción 86
Propiedades de patrón 87
Configuración predeterminada 88
Arquitectura 89
GitHub 90
aws-cloudfront-mediastore 90
Información general 90
Inicializador 91
Patrón de construcción 91
Propiedades de patrón 92
Configuración predeterminada 93

Arquitectura 94
GitHub 94
aws-cloudfront-s3 95
Información general 95
Inicializador 96
Patrón de construcción 96
Propiedades de patrón 97
Configuración predeterminada 98
Arquitectura 99
GitHub 99
aws-cognito-apigateway-lambda 99
Información general 80
Initializer 101
Patrón de construcción 102
Propiedades de patrón 103
Configuración predeterminada 104
Arquitectura 105
GitHub 105
aws-dynamodb-stream-lambda 105
Información general 106
Initializer 107
Patrón de construcción 107
Propiedades de patrón 108
Lambda function 108
Configuración predeterminada 109
Arquitectura 110
GitHub 110
aws-dynamodb-stream-lambda-elasticsearch-kibana 110
Información general 111
Initializer 112
Patrón de construcción 112
Propiedades de patrón 114
Lambda function 114
Configuración predeterminada 115
Arquitectura 116
GitHub 116

aws-events-rule-kinesisfiremanse-s3 117
Información general 117
Inicializador 118
Patrón de construcción 118
Propiedades de patrón 119
Configuración predeterminada 120
Arquitectura 121
GitHub 122
aws-events-rule-kinesisstreams 122
Información general 122
Initializer 123
Patrón de construcción 123
Patrón Propiedades 124
Configuración predeterminada 125
Arquitectura 125
GitHub 126
aws-events-regla-lambda 126
Información general 126
Inicializador 127
Patrón de construcción 127
Propiedades de patrón 128
Configuración predeterminada 128
Arquitectura 129
GitHub 130
aws-events-rule-sns 130
Información general 130
Initializer 131
Patrón de construcción 132
Propiedades de patrón 133
Configuración predeterminada 133
Arquitectura 134
GitHub 134
aws-events-rule-sqs 134
Información general 135
Inicializador 136
Patrón de construcción 136

Propiedades de patrón 138
Configuración predeterminada 139
Arquitectura 140
GitHub 140
aws-events-rule-step-función 140
Información general 141
Inicializador 142
Patrón de construcción 142
Propiedades de patrón 143
Configuración predeterminada 143
Arquitectura 144
GitHub 144
aws-yot-kinesisfiremanse-s3 144
Información general 145
Inicializador 146
Patrón de construcción 146
Propiedades de patrón 147
Configuración predeterminada 148
Arquitectura 149
GitHub 149
aws-yot-lambda 150
Información general 150
Inicializador 151
Patrón de construcción 151
Propiedades de patrón 152
Configuración predeterminada 152
Arquitectura 153
GitHub 154
aws-iot-lambda-dynamodb 154
Información general 154
Initializer 155
Patrón de construcción 156
Propiedades de patrón 157
Configuración predeterminada 157
Arquitectura 158
GitHub 158

aws-kinesisfiremanse-s3 159
Información general 159
Inicializador 160
Patrón de construcción 160
Propiedades de patrón 161
Configuración predeterminada 162
Arquitectura 163
GitHub 163
aws-kinesisfiremanse-s3-y-kinesisanalytics 163
Información general 164
Inicializador 165
Patrón de construcción 165
Propiedades de patrón 167
Configuración predeterminada 167
Arquitectura 169
GitHub 169
aws-kinesisstreams-gluejob 169
Información general 170
Initializer 171
Patrón de construcción 172
SinkDataStoreProps 174
SinkStoreType 174
Configuración predeterminada 174
Arquitectura 176
GitHub 176
aws-kinesisstreams-kinesisfiremanse-s3 177
Información general 177
Inicializer 178
Patrón de construcción 178
Propiedades de patrón 179
Configuración predeterminada 181
Arquitectura 182
GitHub 182
aws-kinesisstreams-lambda 182
Información general 183
Initializer 184

Patrón de construcción 184
Propiedades de patrón 185
Configuración predeterminada 186
Arquitectura 187
GitHub 187
aws-lambda-dynamodb 187
Información general 188
Initializer 189
Patrón de construcción 189
Propiedades de patrón 193
Configuración predeterminada 194
Arquitectura 195
GitHub 195
aws-lambda-elasticsearch-kibana 195
Información general 196
Initializador 197
Patrón de construcción 197
Propiedades de patrón 198
Lambda function 199
Configuración predeterminada 199
Arquitectura 201
GitHub 201
aws-lambda-s3 202
Información general 202
Inicializador 203
Patrón de construcción 203
Propiedades de patrón 207
Configuración predeterminada 208
Arquitectura 209
GitHub 209
aws-lambda-ssmstringparameter 209
Información general 210
Inicializador 211
Patrón de construcción 211
Propiedades de patrón 216
Configuración predeterminada 216

Arquitectura ... 217
GitHub ... 217
aws-lambda-sagemakerendpoint ... 217
Información general ... 218
Inicializador ... 219
Patrón de construcción ... 219
Propiedades de patrón ... 224
Configuración predeterminada ... 224
Arquitectura ... 226
GitHub ... 226
aws-lambda-secretsmanager ... 226
Información general ... 227
Initializer ... 227
Patrón de construcción ... 228
Propiedades de patrón ... 231
Configuración predeterminada ... 232
Arquitectura ... 233
GitHub ... 233
aws-lambda-sns ... 233
Información general ... 234
Inicializador ... 234
Patrón de construcción ... 235
Propiedades de patrón ... 238
Configuración predeterminada ... 239
Arquitectura ... 240
GitHub ... 240
aws-lambda-sqs ... 240
Información general ... 241
Inicializador ... 241
Patrón de construcción ... 242
Patrón ... 246
Configuración predeterminada ... 247
Arquitectura ... 248
GitHub ... 248
aws-lambda-sqs-lambda ... 248
Información general ... 249

Inicializador 250
Patrón de construcción 250
Propiedades de patrón 253
Configuración predeterminada 253
Arquitectura 254
GitHub 254
aws-lambda-step-función 255
Información general 255
Inicializador 256
Patrón de construcción 256
Propiedades de patrón 257
Configuración predeterminada 258
Arquitectura 259
GitHub 259
aws-s3-lambda 259
Información general 260
Inicializador 261
Patrón de construcción 261
Propiedades de patrón 262
Configuración predeterminada 263
Arquitectura 264
GitHub 264
aws-s3-sqs 264
Información general 265
Initializer 265
Patrón de construcción 266
Propiedades de patrón 268
Configuración predeterminada 269
Arquitectura 270
GitHub 270
función de paso aws-s3 270
Información general 271
Initializer 272
Patrón de construcción 272
Propiedades de patrón 274
Configuración predeterminada 275

Arquitectura ... 276
GitHub ... 276
aws-sns-lambda ... 276
Información general ... 277
Inicializador ... 277
Patrón de construcción ... 278
Propiedades de patrón ... 279
Configuración predeterminada ... 279
Arquitectura ... 280
GitHub ... 280
aws-sns-sqs ... 280
Información general ... 281
Initializador ... 282
Patrón de construcción ... 282
Propiedades de patrón ... 284
Configuración predeterminada ... 285
Arquitectura ... 286
GitHub ... 286
aws-sqs-lambda ... 286
Información general ... 287
Inicializador ... 287
Patrón de construcción ... 288
Propiedades de patrón ... 289
Configuración predeterminada ... 290
Arquitectura ... 291
GitHub ... 291
core ... 291
Propiedades predeterminadas para construcciones CDK de AWS ... 292
Anular las propiedades predeterminadas ... 292
Advertencias de reemplazo de propiedades ... 293
Revisiones del documento ... 294
Avisos ... 299
... ccc

Construcciones de soluciones de AWS

Fecha de publicación: Mayo de 2021(Revisiones del documento)

¿Qué es AWS Solutions Constructs?

AWS Solutions Constructs (Constructs) es una extensión de código abierto delKit de desarrollo de la nube de AWS (AWS CDK)que proporciona patrones multiservicio y bien diseñados para definir rápidamente soluciones en código con el fin de crear una infraestructura predecible y repetible. El objetivo es acelerar la experiencia de los desarrolladores para crear soluciones de cualquier tamaño utilizando definiciones basadas en patrones para su arquitectura.

Utilice AWS Solutions Constructs para definir sus soluciones en un lenguaje de programación familiar. AWS Solutions Constructs admite TypeScript, JavaScript, Python y Java en este momento.

Para explorar el catálogo completo de patrones de AWS Solutions Constructs,Haga clic aquí.

¿Por qué utilizar AWS Solutions Constructs?

Con la tasa de innovación de los proveedores de nube, conocer y comprender las mejores prácticas y garantizar que se implementen correctamente en toda la solución puede ser desalentador. Los componentes fijos le permiten combinar patrones preconstruidos y bien diseñados y casos de uso que realizan acciones comunes utilizando servicios en la nube de una manera escalable y segura. Dado que Constructs proporciona una biblioteca para lenguajes de programación modernos, puede aplicar habilidades de desarrollo existentes y herramientas familiares a la tarea de crear una infraestructura de nube bien diseñada para sus soluciones.

Otras ventajas de AWS Solutions Constructs incluyen:

- Se basa en el marco de desarrollo de software de código abierto del kit de desarrollo de la nube de AWS (AWS CDK).
- Use lógica (sentencias if, for-loops, etc.) al definir la infraestructura de su solución.
- Utilice técnicas orientadas a objetos para crear un modelo de su sistema.
- Defina abstracciones de alto nivel, compártelas y publíquelas en su equipo, empresa o comunidad.
- Organice sus soluciones en módulos lógicos.
- Comparte y reutiliza tu solución como biblioteca.
- Pruebe su código de infraestructura utilizando protocolos estándar de la industria.

- Utilice el flujo de trabajo de revisión de código existente.

El objetivo de AWS Solutions Constructs es reducir la complejidad y la lógica de pegamento necesarias al integrar patrones comunes bien diseñados para alcanzar los objetivos de su solución en AWS.

Introducción a AWS Solutions

En este tema se describe cómo instalar y configurar AWS Cloud Development Kit (AWS CDK), AWS Solutions Constructs y crear su primera aplicación de AWS CDK utilizando patrones de AWS Solutions Constructs.

> **Note**
>
> Las versiones de CDK de AWS son compatibles con AWS Solutions Constructs ≥ 1.46.0.

> **Tip**
>
> ¿Quieres cavar más profundo? Pruebe elTaller de CDKpara un recorrido más profundo por un proyecto del mundo real.

> **Tip**
>
> Para obtener más información acerca de cómo empezar a utilizar el Cloud Development Kit de AWS (CDK de AWS), consulte laGuía para desarrolladores de CDK de AWS.

Prerequisites

AWS Solutions Constructs se basa en el CDK de AWS, por lo que debe instalar Node.js (>= 10.3.0), incluso aquellos que trabajan en idiomas distintos de TypeScript o JavaScript. Esto se debe a que elCDK DE AWSy AWS Solutions Constructs se desarrollan en TypeScript y se ejecutan en Node.js. Los enlaces para otros idiomas compatibles utilizan este back-end y conjunto de herramientas.

Debe proporcionar sus credenciales y una región de AWS para utilizar la CLI de AWS CDK, tal y como se describe en Especificar sus credenciales y región.

Otros requisitos previos dependen de su lenguaje de desarrollo, de la siguiente manera.

Idioma	Requisitos previos
P	Python >= 3.6
TS T t	TypeScript >= 2.7
Ja	Java >= 1.8

Instalación del CDK de AWS

Para instalar y configurar el CDK de AWS, consulte la Guía para desarrolladores de CDK de AWS:Instalación del CDK de AWS.

Uso de componentes de soluciones de AWS

El flujo de trabajo típico para crear una nueva aplicación cuando se trabaja con AWS Solutions Constructs sigue el mismo enfoque que el CDK de AWS.

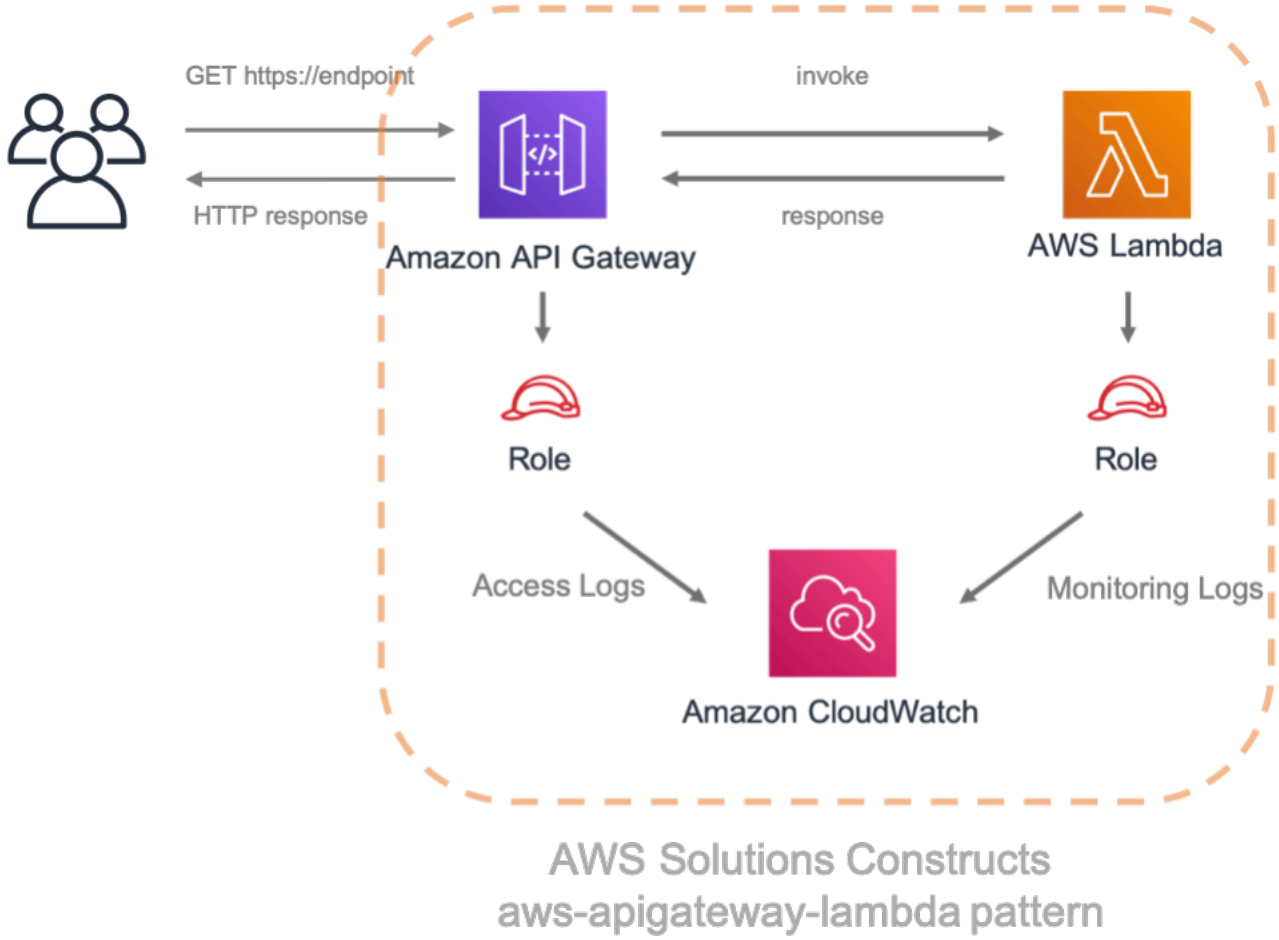
1. Cree el directorio de aplicaciones.
2. Inicialización de la aplicación.
3. Agregue las dependencias de patrones de AWS Solutions Constructs.
4. Añade código adicional a la aplicación.
5. Compile la aplicación, si es necesario.
6. Implemente los recursos definidos en la aplicación.
7. Probar la aplicación.

Si hay algún problema, realice un bucle a través de modificar, compilar (si es necesario), implementar y probar de nuevo.

Tutorial - Parte 1

> **Note**
>
> Construcciones de AWS Solutions se admite en las versiones de AWS CDK ≥ 1.46.0.

Este tutorial le guía a través de cómo crear e implementar una aplicación sencilla de AWS CDK «Hello Constructs» que utiliza un patrón de AWS Solutions Constructs, desde la inicialización del proyecto hasta la implementación de la plantilla de AWS CloudFormation resultante. La aplicación Hello Constructs creará la siguiente solución simple:



Construcciones de Hello

Comencemos a crear nuestra primera aplicación AWS CDK mediante el desarrollo basado en patrones.

> **Note**
>
> Esta es una modificación de muestra de`Hello CDK!`desde las[Taller de CDK](). Si es la primera vez que utiliza AWS CDK, le recomendamos comenzar con este taller para obtener un tutorial práctico y cómo aprovechar el CDK en la creación de un proyecto del mundo real.

Creación del directorio de aplicaciones e inicialización del CDK de AWS

Cree un directorio para su aplicación CDK y, a continuación, cree una aplicación de AWS CDK en ese directorio.

TypeScript

```
mkdir hello-constructs
cd hello-constructs
cdk init --language typescript
```

Python

```
mkdir hello-constructs
cd hello-constructs
cdk init --language python
```

> **Tip**
>
> Ahora es un buen momento para abrir el proyecto en su IDE favorito y explorar. Para obtener más información acerca de la estructura del proyecto, seleccione el vínculo apropiado:
>
> - TypeScript
> - Python

Actualizar dependencias de la base del proyecto

> **Warning**
>
> Para garantizar una funcionalidad adecuada, AWS Solutions Constructs y los paquetes CDK de AWS deben utilizar el mismo número de versión dentro del proyecto. Por ejemplo, si utiliza AWS Solutions Constructs v.1.52.0, también debe utilizar AWS CDK v.1.52.0.

> **Tip**
>
> Tome nota de la versión más reciente de AWS Solutions Constructs y aplique ese número de versión al`VERSION_NUMBER`en los pasos siguientes (para los paquetes de AWS Solutions Constructs y AWS CDK). Para comprobar todas las versiones públicas de la biblioteca de componentes fijos,Click here.

TypeScript

Edite el`package.json`con la siguiente información:

```
"devDependencies": {
  "@aws-cdk/assert": "VERSION_NUMBER",
  "@types/jest": "^24.0.22",
  "@types/node": "10.17.5",
  "jest": "^24.9.0",
  "ts-jest": "^24.1.0",
  "aws-cdk": "VERSION_NUMBER",
  "ts-node": "^8.1.0",
  "typescript": "~3.7.2"
},
"dependencies": {
  "@aws-cdk/core": "VERSION_NUMBER",
  "source-map-support": "^0.5.16"
}
```

Python

Edite el`setup.py`con la siguiente información:

```
install_requires=[
    "aws-cdk.core==VERSION_NUMBER",
],
```

Instale las dependencias base de proyectos.

TypeScript

```
npm install
```

Python

```
source .venv/bin/activate
pip install -r requirements.txt
```

Cree y ejecute la aplicación y confirme que cree una pila vacía.

TypeScript

```
npm run build
cdk synth
```

Python

```
cdk synth
```

Debería ver una pila como la siguiente, donde`CDK-VERSION`es la versión del CDK. (Su salida puede diferir ligeramente de lo que se muestra aquí).

TypeScript

```
Resources:
  CDKMetadata:
    Type: AWS::CDK::Metadata
    Properties:
          Modules: aws-cdk=CDK-VERSION,@aws-cdk/core=VERSION_NUMBER,@aws-cdk/cx-
api=VERSION_NUMBER,jsii-runtime=node.js/10.17.0
```

Python

```
Resources:
  CDKMetadata:
    Type: AWS::CDK::Metadata
    Properties:
          Modules: aws-cdk=CDK-VERSION,@aws-cdk/core=VERSION_NUMBER,@aws-cdk/cx-
api=VERSION_NUMBER,jsii-runtime=Python/3.7.7
```

Código del controlador Lambda

Comenzaremos con el código de controlador de AWS Lambda.

Creación de un directorio de`lambda`En la raíz del árbol de proyecto.

TypeScript

Añadir un archivo denominado`lambda/hello.js`con el siguiente contenido:

```
exports.handler = async function(event) {
  console.log("request:", JSON.stringify(event, null, 2));
  return {
    statusCode: 200,
    headers: { "Content-Type": "text/plain" },
    body: `Hello, AWS Solutions Constructs! You've hit ${event.path}\n`
  };
};
```

Python

Añadir un archivo denominado`lambda/hello.py`con el siguiente contenido:

```
import json

def handler(event, context):
    print('request: {}'.format(json.dumps(event)))
    return {
        'statusCode': 200,
        'headers': {
            'Content-Type': 'text/plain'
        },
        'body': 'Hello, CDK! You have hit {}\n'.format(event['path'])
    }
```

Esta es una simple función de Lambda que devuelve el texto «Hola, Constructs! Has presionado [ruta URL]». La salida de la función también incluye el código de estado HTTP y los encabezados HTTP. Estos son utilizados por API Gateway para formular la respuesta HTTP al usuario.

Este Lambda se proporciona en JavaScript. Para obtener más información sobre cómo escribir funciones de Lambda en el idioma de su elección, consulte laDocumentación de AWS Lambda.

Instalar las dependencias de AWS CDK y AWS Solutions Constructs

Los componentes fijos de soluciones de AWS se suministran con una amplia biblioteca de componentes fijos. La biblioteca está dividida en módulos, uno para cada patrón bien diseñado. Por ejemplo, si desea definir una API de descanso de Amazon API Gateway para una función de AWS Lambda, necesitaremos utilizar la`aws-apigateway-lambda`Biblioteca de patrones.

También tenemos que añadir la biblioteca de construcción de AWS Lambda y Amazon API Gateway desde el CDK de AWS.

Instale el módulo AWS Lambda y todas sus dependencias en nuestro proyecto:

> **Note**
>
> Recuerde sustituir la versión correcta y coincidente que se utilizará tanto para AWS Solutions Constructs como para AWS CDK en el`VERSION_NUMBER`campos de marcador de posición para cada comando. Las versiones incorrectas entre los paquetes pueden causar errores.

TypeScript

```
npm install -s @aws-cdk/aws-lambda@VERSION_NUMBER
```

Python

```
pip install aws_cdk.aws_lambda==VERSION_NUMBER
```

A continuación, instale el módulo Amazon API Gateway y todas sus dependencias en nuestro proyecto:

TypeScript

```
npm install -s @aws-cdk/aws-apigateway@VERSION_NUMBER
```

Python

```
pip install aws_cdk.aws_apigateway==VERSION_NUMBER
```

Por último, instale los constructos de soluciones de AWS`aws-apigateway-lambda`y todas sus dependencias en nuestro proyecto:

TypeScript

```
npm install -s @aws-solutions-constructs/aws-apigateway-lambda@VERSION_NUMBER
```

Python

```
pip install aws_solutions_constructs.aws_apigateway_lambda==VERSION_NUMBER
```

Agregar un patrón de Amazon API Gateway/AWS Lambda a su pila

Ahora definamos el patrón de AWS Solutions Constructs para implementar una Amazon API Gateway con un proxy de AWS Lambda.

TypeScript

Edite el archivo`lib/hello-constructs.ts`con lo siguiente:

```
import * as cdk from '@aws-cdk/core';
import * as lambda from '@aws-cdk/aws-lambda';
import * as api from '@aws-cdk/aws-apigateway';
import { ApiGatewayToLambda, ApiGatewayToLambdaProps } from '@aws-solutions-
constructs/aws-apigateway-lambda';

export class HelloConstructsStack extends cdk.Stack {
  constructor(scope: cdk.Construct, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    // The code that defines your stack goes here
    const api_lambda_props: ApiGatewayToLambdaProps = {
      lambdaFunctionProps: {
        code: lambda.Code.fromAsset('lambda'),
        runtime: lambda.Runtime.NODEJS_12_X,
        handler: 'hello.handler'
      },
```

```
      apiGatewayProps: {
        defaultMethodOptions: {
          authorizationType: api.AuthorizationType.NONE
        }
      }
    };

    new ApiGatewayToLambda(this, 'ApiGatewayToLambda', api_lambda_props);
  }
}
```

Python

Edite el archivo`hello_constructs/hello_constructs_stack.py`con lo siguiente:

```
from aws_cdk import (
    aws_lambda as _lambda,
    aws_apigateway as apigw,
    core,
)

from aws_solutions_constructs import (
    aws_apigateway_lambda as apigw_lambda
)

class HelloConstructsStack(core.Stack):

    def __init__(self, scope: core.Construct, id: str, **kwargs) -> None:
        super().__init__(scope, id, **kwargs)

        # The code that defines your stack goes here

        apigw_lambda.ApiGatewayToLambda(
            self, 'ApiGatewayToLambda',
            lambda_function_props=_lambda.FunctionProps(
                runtime=_lambda.Runtime.PYTHON_3_7,
                code=_lambda.Code.asset('lambda'),
                handler='hello.handler',
            ),
            api_gateway_props=apigw.RestApiProps(
                default_method_options=apigw.MethodOptions(
                    authorization_type=apigw.AuthorizationType.NONE
```

```
                )
            )
        )
```

Eso es todo. Esto es todo lo que necesita hacer para definir una API Gateway que proxie todas las solicitudes a una función de AWS Lambda. Vamos a comparar nuestra nueva pila con la original:

TypeScript

```
npm run build
cdk diff
```

Python

```
cdk diff
```

La salida debe tener el siguiente aspecto:

```
Stack HelloConstructsStack
IAM Statement Changes
###########################################################################################
#   # Resource                    # Effect # Action                      # Principal
                # Condition                   #
###########################################################################################
# + # ${LambdaFunction.Arn}       # Allow  # lambda:InvokeFunction       #
 Service:apigateway.amazonaw # "ArnLike": {                 #
#   #                             #        #                             # s.com
                #   "AWS:SourceArn": "arn:${AW #
#   #                             #        #                             #
                # S::Partition}:execute-api:${ #
#   #                             #        #                             #
                # AWS::Region}:${AWS::AccountI #
#   #                             #        #                             #
                # d}:${RestApi0C43BF4B}/${Rest #
```

```
#   #                               #        #                               #
                # Api/DeploymentStage.prod}/*/ #
#   #                               #        #                               #
                # {proxy+}"                    #
#   #                               #        #                               #
                # }                            #
# + # ${LambdaFunction.Arn}       # Allow  # lambda:InvokeFunction       #
 Service:apigateway.amazonaw # "ArnLike": {                 #
#   #                               #        #                               # s.com
                #   "AWS:SourceArn": "arn:${AW #
#   #                               #        #                               #
                # S::Partition}:execute-api:${ #
#   #                               #        #                               #
                # AWS::Region}:${AWS::AccountI #
#   #                               #        #                               #
                # d}:${RestApi0C43BF4B}/test-i #
#   #                               #        #                               #
                # nvoke-stage/*/{proxy+}"      #
#   #                               #        #                               #
                # }                            #
# + # ${LambdaFunction.Arn}       # Allow  # lambda:InvokeFunction       #
 Service:apigateway.amazonaw # "ArnLike": {                 #
#   #                               #        #                               # s.com
                #   "AWS:SourceArn": "arn:${AW #
#   #                               #        #                               #
                # S::Partition}:execute-api:${ #
#   #                               #        #                               #
                # AWS::Region}:${AWS::AccountI #
#   #                               #        #                               #
                # d}:${RestApi0C43BF4B}/${Rest #
#   #                               #        #                               #
                # Api/DeploymentStage.prod}/*/ #
#   #                               #        #                               #
                # "                            #
#   #                               #        #                               #
                # }                            #
# + # ${LambdaFunction.Arn}       # Allow  # lambda:InvokeFunction       #
 Service:apigateway.amazonaw # "ArnLike": {                 #
#   #                               #        #                               # s.com
                #   "AWS:SourceArn": "arn:${AW #
#   #                               #        #                               #
                # S::Partition}:execute-api:${ #
#   #                               #        #                               #
                # AWS::Region}:${AWS::AccountI #
```

```
#   #                               #        #                              #
                # d}:${RestApi0C43BF4B}/test-i #
#   #                               #        #                              #
                # nvoke-stage/*/"              #
#   #                               #        #                              #
                # }                            #
##########################################################################################
# + # ${LambdaFunctionServiceRole # Allow  # sts:AssumeRole              #
 Service:lambda.amazonaws.co #                                #
#   # .Arn}                       #        #                              # m
                #                              #
##########################################################################################
# + # ${LambdaRestApiCloudWatchRo # Allow  # sts:AssumeRole              #
 Service:apigateway.amazonaw #                                #
#   # le.Arn}                     #        #                              # s.com
                #                              #
##########################################################################################
# + # arn:aws:logs:${AWS::Region} # Allow  # logs:CreateLogGroup         # AWS:
${LambdaRestApiCloudWat #                                 #
#   # :${AWS::AccountId}:*        #        # logs:CreateLogStream        # chRole}
                #                              #
#   #                               #        # logs:DescribeLogGroups      #
                #                              #
#   #                               #        # logs:DescribeLogStreams     #
                #                              #
#   #                               #        # logs:FilterLogEvents        #
                #                              #
#   #                               #        # logs:GetLogEvents           #
                #                              #
#   #                               #        # logs:PutLogEvents           #
                #                              #
##########################################################################################
# + # arn:aws:logs:${AWS::Region} # Allow  # logs:CreateLogGroup         # AWS:
${LambdaFunctionService #                                 #
#   # :${AWS::AccountId}:log-grou #        # logs:CreateLogStream        # Role}
                #                              #
#   # p:/aws/lambda/*             #        # logs:PutLogEvents           #
                #                              #
##########################################################################################
(NOTE: There may be security-related changes not in this list. See https://github.com/
aws/aws-cdk/issues/1299)

Parameters
```

```
[+] Parameter AssetParameters/
ba91444ebd644d9419e8cfee417f3aaa728507dd428788a2fc40574646c4340a/S3Bucket
 AssetParametersba91444ebd644d9419e8cfee417f3aaa728507dd428788a2fc40574646c4340aS3Bucket9780A3B
 {"Type":"String","Description":"S3 bucket for asset
 \"ba91444ebd644d9419e8cfee417f3aaa728507dd428788a2fc40574646c4340a\""}
[+] Parameter AssetParameters/
ba91444ebd644d9419e8cfee417f3aaa728507dd428788a2fc40574646c4340a/S3VersionKey
 AssetParametersba91444ebd644d9419e8cfee417f3aaa728507dd428788a2fc40574646c4340aS3VersionKey37F
 {"Type":"String","Description":"S3 key for asset version
 \"ba91444ebd644d9419e8cfee417f3aaa728507dd428788a2fc40574646c4340a\""}
[+] Parameter AssetParameters/
ba91444ebd644d9419e8cfee417f3aaa728507dd428788a2fc40574646c4340a/ArtifactHash
 AssetParametersba91444ebd644d9419e8cfee417f3aaa728507dd428788a2fc40574646c4340aArtifactHash801
 {"Type":"String","Description":"Artifact hash for asset
 \"ba91444ebd644d9419e8cfee417f3aaa728507dd428788a2fc40574646c4340a\""}

Conditions
[+] Condition CDKMetadataAvailable: {"Fn::Or":[{"Fn::Or":[{"Fn::Equals":
[{"Ref":"AWS::Region"},"ap-east-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"ap-
northeast-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"ap-northeast-2"]},{"Fn::Equals":
[{"Ref":"AWS::Region"},"ap-south-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"ap-
southeast-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"ap-southeast-2"]},{"Fn::Equals":
[{"Ref":"AWS::Region"},"ca-central-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"cn-
north-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"cn-northwest-1"]},
{"Fn::Equals":[{"Ref":"AWS::Region"},"eu-central-1"]}]},{"Fn::Or":[{"Fn::Equals":
[{"Ref":"AWS::Region"},"eu-north-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"eu-
west-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"eu-west-2"]},{"Fn::Equals":
[{"Ref":"AWS::Region"},"eu-west-3"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"me-
south-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"sa-east-1"]},{"Fn::Equals":
[{"Ref":"AWS::Region"},"us-east-1"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"us-
east-2"]},{"Fn::Equals":[{"Ref":"AWS::Region"},"us-west-1"]},{"Fn::Equals":
[{"Ref":"AWS::Region"},"us-west-2"]}]}]}

Resources
[+] AWS::Logs::LogGroup ApiGatewayToLambda/ApiAccessLogGroup
 ApiGatewayToLambdaApiAccessLogGroupE2B41502
[+] AWS::IAM::Role LambdaFunctionServiceRole LambdaFunctionServiceRole0C4CDE0B
[+] AWS::Lambda::Function LambdaFunction LambdaFunctionBF21E41F
[+] AWS::ApiGateway::RestApi RestApi RestApi0C43BF4B
[+] AWS::ApiGateway::Deployment RestApi/Deployment
 RestApiDeployment180EC503d2c6df3c8dc8b7193b98c1a0bff4e677
[+] AWS::ApiGateway::Stage RestApi/DeploymentStage.prod
 RestApiDeploymentStageprod3855DE66
[+] AWS::ApiGateway::Resource RestApi/Default/{proxy+} RestApiproxyC95856DD
```

```
[+] AWS::Lambda::Permission RestApi/Default/{proxy+}/ANY/
ApiPermission.HelloConstructsStackRestApiFDB18C2E.ANY..{proxy+}
 RestApiproxyANYApiPermissionHelloConstructsStackRestApiFDB18C2EANYproxyE43D39B3
[+] AWS::Lambda::Permission RestApi/Default/{proxy+}/ANY/
ApiPermission.Test.HelloConstructsStackRestApiFDB18C2E.ANY..{proxy+}
 RestApiproxyANYApiPermissionTestHelloConstructsStackRestApiFDB18C2EANYproxy0B23CDC7
[+] AWS::ApiGateway::Method RestApi/Default/{proxy+}/ANY RestApiproxyANY1786B242
[+] AWS::Lambda::Permission RestApi/Default/ANY/
ApiPermission.HelloConstructsStackRestApiFDB18C2E.ANY..
 RestApiANYApiPermissionHelloConstructsStackRestApiFDB18C2EANY5684C1E6
[+] AWS::Lambda::Permission RestApi/Default/ANY/
ApiPermission.Test.HelloConstructsStackRestApiFDB18C2E.ANY..
 RestApiANYApiPermissionTestHelloConstructsStackRestApiFDB18C2EANY81DBDF56
[+] AWS::ApiGateway::Method RestApi/Default/ANY RestApiANYA7C1DC94
[+] AWS::ApiGateway::UsagePlan RestApi/UsagePlan RestApiUsagePlan6E1C537A
[+] AWS::Logs::LogGroup ApiAccessLogGroup ApiAccessLogGroupCEA70788
[+] AWS::IAM::Role LambdaRestApiCloudWatchRole LambdaRestApiCloudWatchRoleF339D4E6
[+] AWS::ApiGateway::Account LambdaRestApiAccount LambdaRestApiAccount

Outputs
[+] Output RestApi/Endpoint RestApiEndpoint0551178A: {"Value":{"Fn::Join":["",
["https://",{"Ref":"RestApi0C43BF4B"},".execute-api.",{"Ref":"AWS::Region"},".",
{"Ref":"AWS::URLSuffix"},"/",{"Ref":"RestApiDeploymentStageprod3855DE66"},"/"]]}}
```

Eso es bonito. Este sencillo ejemplo con un patrón bien diseñado de AWS Solutions Constructs añadió 21 nuevos recursos a su pila.

Implementar cdk

> **Tip**
>
> Antes de poder implementar su primera aplicación de AWS CDK que contenga una función Lambda, debe iniciar su entorno de AWS. Esto crea un depósito provisional que el CDK de AWS utiliza para implementar pilas que contienen activos. Si esta es la primera vez que utiliza AWS CDK para implementar activos, deberá ejecutar el`cdk bootstrap`para implementar la pila del kit de herramientas CDK en su entorno de AWS.

¿Listo para la implementación?

```
cdk deploy
```

Salida de la pila

Cuando se haya completado la implementación, notará esta línea:

```
Outputs:
HelloConstructsStack.RestApiEndpoint0551178A = https://xxxxxxxxxx.execute-api.us-
east-1.amazonaws.com/prod/
```

Se trata de una salida de pila que se agrega automáticamente mediante el patrón AWS Solutions Constructs e incluye la URL del extremo API Gateway.

Probar su aplicación

Vamos a tratar de llegar a este punto final con`curl`. Copie la URL y ejecute (es probable que su prefijo y Región sean diferentes).

```
curl https://xxxxxxxxxx.execute-api.us-east-1.amazonaws.com/prod/
```

La salida debe tener el siguiente aspecto:

```
Hello, AWS Solutions Constructs! You've hit /
```

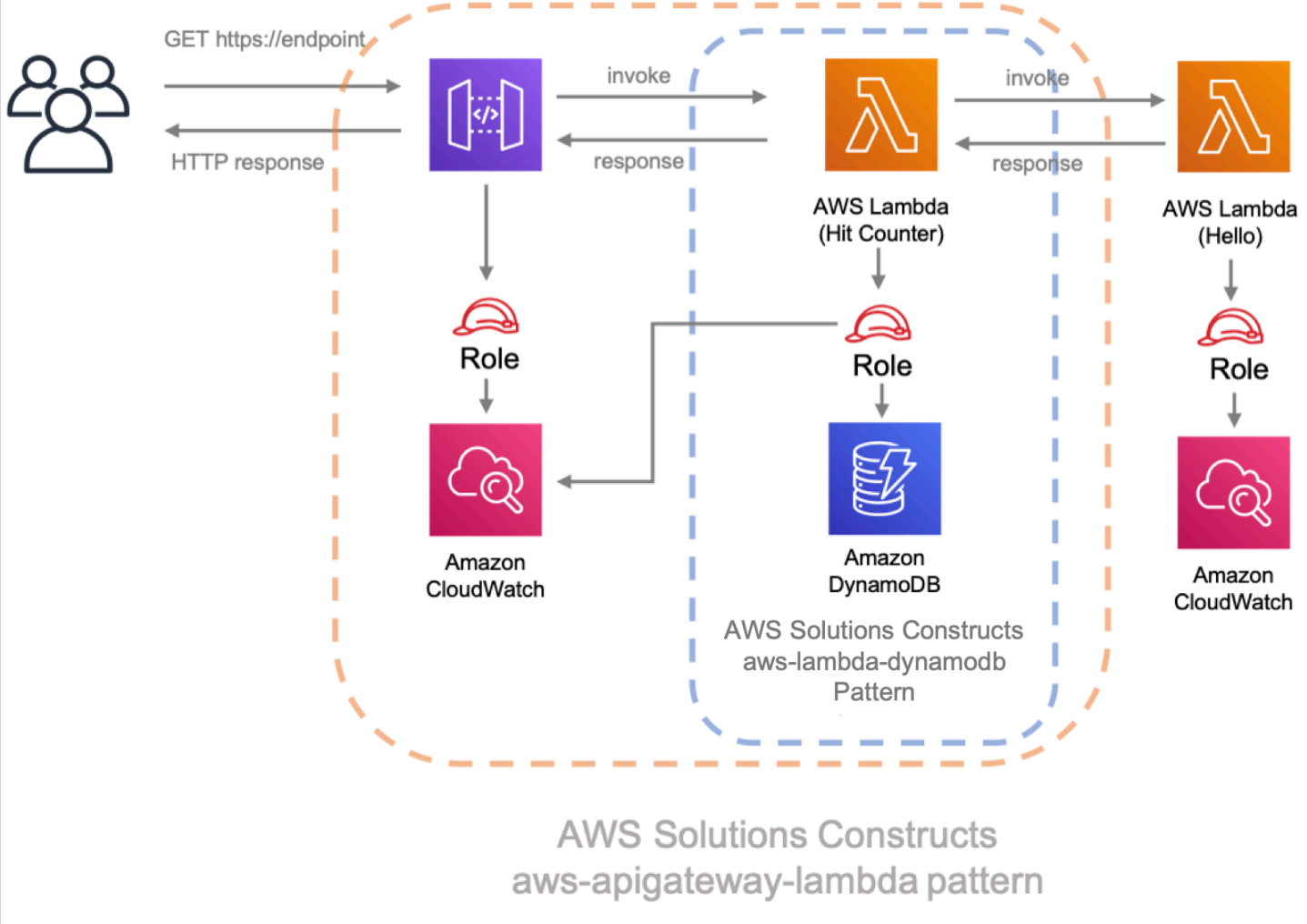
Si esta es la salida que recibiste, ¡tu aplicación funciona!

Tutorial - Parte 2

> **Note**
>
> AWS Solutions Constructs es compatible con las versiones de AWS CDK ≥ 1.46.0.

Este tutorial le guiará a través de cómo modificar la aplicación «Hello Constructs» creada enParte 1. Nuestra modificación agregará un contador de visitas de sitios utilizando el patrón AWS Lambda

al DynamoDB de AWS Solutions Constructs. La modificación de la aplicación Hello Constructs dará como resultado la siguiente solución:



Código de Hit Counter Lambda

Comencemos escribiendo el código para la función Hit Counter AWS Lambda. Esta función:

- aumentar un contador relacionado con la ruta de la API en una tabla de Amazon DynamoDB,
- invocar la función Hello AWS Lambda descendente,
- y devolver la respuesta al usuario final.

TypeScript

Añadir un archivo denominado`lambda/hitcounter.js`con el siguiente contenido:

```
  const { DynamoDB, Lambda } = require('aws-sdk');

exports.handler = async function(event) {
  console.log("request:", JSON.stringify(event, undefined, 2));

  // create AWS SDK clients
  const dynamo = new DynamoDB();
  const lambda = new Lambda();

  // update dynamo entry for "path" with hits++
  await dynamo.updateItem({
    TableName: process.env.DDB_TABLE_NAME,
    Key: { path: { S: event.path } },
    UpdateExpression: 'ADD hits :incr',
    ExpressionAttributeValues: { ':incr': { N: '1' } }
  }).promise();

  // call downstream function and capture response
  const resp = await lambda.invoke({
    FunctionName: process.env.DOWNSTREAM_FUNCTION_NAME,
    Payload: JSON.stringify(event)
  }).promise();

  console.log('downstream response:', JSON.stringify(resp, undefined, 2));

  // return response back to upstream caller
  return JSON.parse(resp.Payload);
};
```

Python

Añadir un archivo denominado`lambda/hitcounter.py`con el siguiente contenido:

```
import json
import os
import boto3

ddb = boto3.resource('dynamodb')
table = ddb.Table(os.environ['DDB_TABLE_NAME'])
_lambda = boto3.client('lambda')
```

```
def handler(event, context):
    print('request: {}'.format(json.dumps(event)))
    table.update_item(
        Key={'path': event['path']},
        UpdateExpression='ADD hits :incr',
        ExpressionAttributeValues={':incr': 1}
    )

    resp = _lambda.invoke(
        FunctionName=os.environ['DOWNSTREAM_FUNCTION_NAME'],
        Payload=json.dumps(event),
    )

    body = resp['Payload'].read()

    print('downstream response: {}'.format(body))
    return json.loads(body)
```

Instalar las nuevas dependencias

> **Note**
>
> Recuerde sustituir la versión correcta y coincidente que se utilizará tanto para AWS Solutions Constructs como para AWS CDK en el`VERSION_NUMBER`campos de marcador de posición para cada comando. Debe ser idéntico al número de versión utilizado para las dependencias en la primera parte de este tutorial. Las versiones incorrectas entre los paquetes pueden causar errores.

Como de costumbre, primero tenemos que instalar las dependencias que necesitamos para la actualización de nuestra solución. En primer lugar, tenemos que instalar la biblioteca de construcción de DynamoDB:

TypeScript

```
npm install -s @aws-cdk/aws-dynamodb@VERSION_NUMBER
```

Python

```
pip install aws_cdk.aws_dynamodb==VERSION_NUMBER
```

Por último, instale los constructos de soluciones de AWS`aws-lambda-dynamodb`y todas sus dependencias en nuestro proyecto:

TypeScript

```
npm install -s @aws-solutions-constructs/aws-lambda-dynamodb@VERSION_NUMBER
```

Python

```
pip install aws_solutions_constructs.aws_lambda_dynamodb==VERSION_NUMBER
```

Defina los recursos

Ahora, actualicemos nuestro código de pila para acomodar nuestra nueva arquitectura.

Primero, vamos a importar nuestras nuevas dependencias y mover la función «Hola» fuera del`aws-apigateway-lambda`que hemos creado en la parte 1.

TypeScript

Editar el archivo`lib/hello-constructs.ts`con lo siguiente:

```
import * as cdk from '@aws-cdk/core';
import * as lambda from '@aws-cdk/aws-lambda';
import * as api from '@aws-cdk/aws-apigateway';
import * as dynamodb from '@aws-cdk/aws-dynamodb';
import { ApiGatewayToLambda, ApiGatewayToLambdaProps } from '@aws-solutions-
constructs/aws-apigateway-lambda';
```

```
import { LambdaToDynamoDB, LambdaToDynamoDBProps } from '@aws-solutions-constructs/
aws-lambda-dynamodb';

export class HelloConstructsStack extends cdk.Stack {
  constructor(scope: cdk.Construct, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    // The code that defines your stack goes here

    const helloFunc = new lambda.Function(this, 'HelloHandler', {
      runtime: lambda.Runtime.NODEJS_12_X,
      code: lambda.Code.fromAsset('lambda'),
      handler: 'hello.handler'
    });

    const api_lambda_props: ApiGatewayToLambdaProps = {
      lambdaFunctionProps: {
        code: lambda.Code.fromAsset('lambda'),
        runtime: lambda.Runtime.NODEJS_12_X,
        handler: 'hello.handler'
      },
      apiGatewayProps: {
        defaultMethodOptions: {
          authorizationType: api.AuthorizationType.NONE
        }
      }
    };

    new ApiGatewayToLambda(this, 'ApiGatewayToLambda', api_lambda_props);
  }
}
```

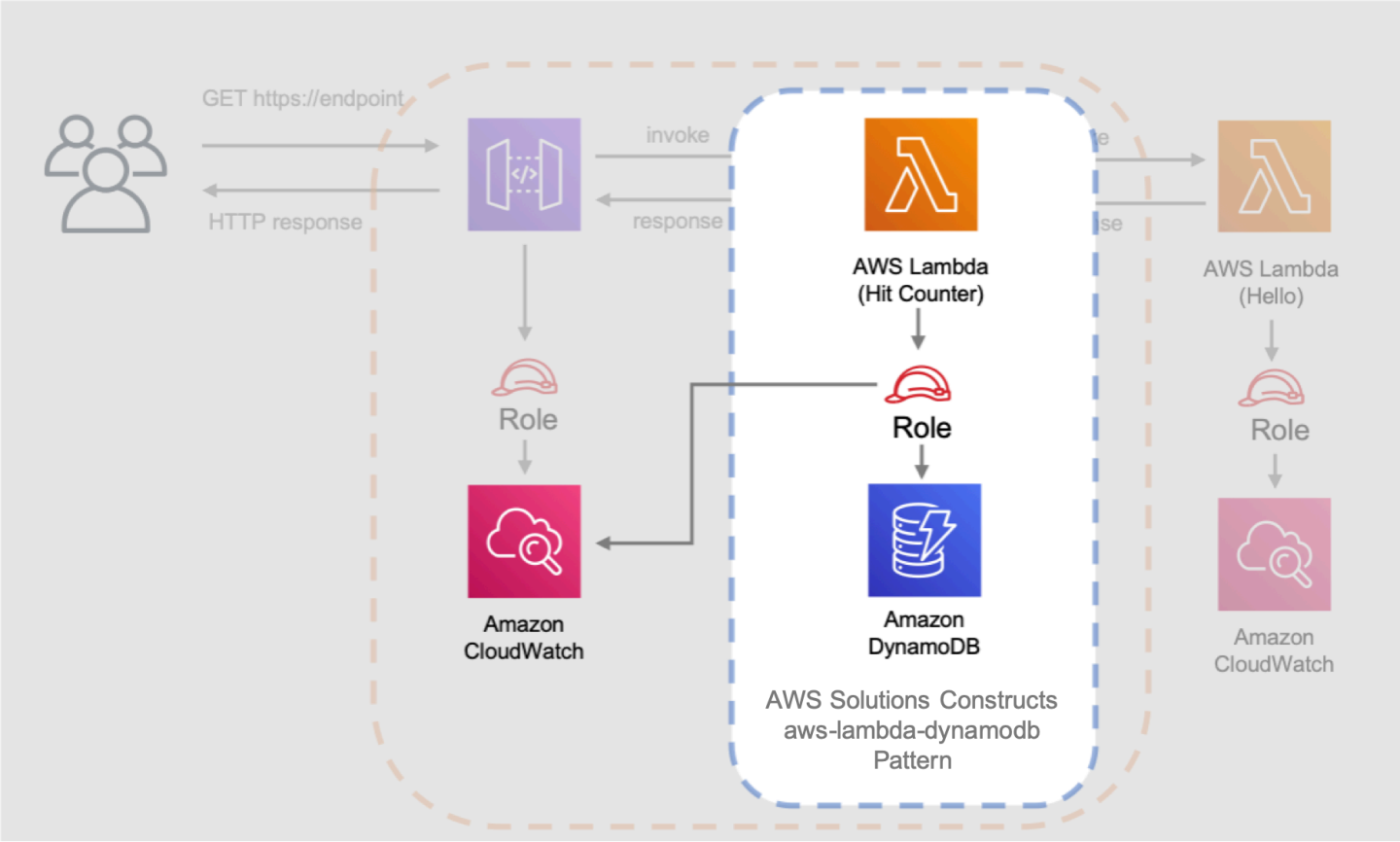
Python

Editar el archivo`hello_constructs/hello_constructs_stack.py`con lo siguiente:

```
from aws_cdk import (
    aws_lambda as _lambda,
    aws_apigateway as apigw,
    aws_dynamodb as ddb,
    core,
)
```

```
from aws_solutions_constructs import (
    aws_apigateway_lambda as apigw_lambda,
    aws_lambda_dynamodb as lambda_ddb
)

class HelloConstructsStack(core.Stack):

    def __init__(self, scope: core.Construct, id: str, **kwargs) -> None:
        super().__init__(scope, id, **kwargs)

        # The code that defines your stack goes here

        self._handler = _lambda.Function(
          self, 'HelloHandler',
          runtime=_lambda.Runtime.PYTHON_3_7,
          handler='hello.handler',
          code=_lambda.Code.asset('lambda'),
        )

        apigw_lambda.ApiGatewayToLambda(
            self, 'ApiGatewayToLambda',
            lambda_function_props=_lambda.FunctionProps(
                runtime=_lambda.Runtime.PYTHON_3_7,
                code=_lambda.Code.asset('lambda'),
                handler='hello.handler',
            ),
            api_gateway_props=apigw.RestApiProps(
                default_method_options=apigw.MethodOptions(
                    authorization_type=apigw.AuthorizationType.NONE
                )
            )
        )
```

A continuación, vamos a añadir el`aws-lambda-dynamodb`para construir el servicio de contador de visitas para nuestra arquitectura actualizada.



AWS Solutions Constructs
aws-apigateway-lambda pattern

En la siguiente actualización se definen las propiedades de la`aws-lambda-dynamodb`mediante la definición de la función AWS Lambda con el controlador de contador de visitas. Además, la tabla de Amazon DynamoDB se define con un nombre`Hitsy` una clave de partición de`path`.

TypeScript

Editar el archivo`lib/hello-constructs.ts`con lo siguiente:

```
import * as cdk from '@aws-cdk/core';
import * as lambda from '@aws-cdk/aws-lambda';
import * as api from '@aws-cdk/aws-apigateway';
import * as dynamodb from '@aws-cdk/aws-dynamodb';
import { ApiGatewayToLambda, ApiGatewayToLambdaProps } from '@aws-solutions-
constructs/aws-apigateway-lambda';
import { LambdaToDynamoDB, LambdaToDynamoDBProps } from '@aws-solutions-constructs/
aws-lambda-dynamodb';
```

```
export class HelloConstructsStack extends cdk.Stack {
  constructor(scope: cdk.Construct, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    // The code that defines your stack goes here

    const helloFunc = new lambda.Function(this, 'HelloHandler', {
      runtime: lambda.Runtime.NODEJS_12_X,
      code: lambda.Code.fromAsset('lambda'),
      handler: 'hello.handler'
    });

    // hit counter, aws-lambda-dynamodb pattern
    const lambda_ddb_props: LambdaToDynamoDBProps = {
      lambdaFunctionProps: {
          code: lambda.Code.asset(`lambda`),
          runtime: lambda.Runtime.NODEJS_12_X,
          handler: 'hitcounter.handler',
          environment: {
              DOWNSTREAM_FUNCTION_NAME: helloFunc.functionName
          }
      },
      dynamoTableProps: {
          tableName: 'Hits',
          partitionKey: { name: 'path', type: dynamodb.AttributeType.STRING }
      }
    };

    const hitcounter = new LambdaToDynamoDB(this, 'LambdaToDynamoDB',
 lambda_ddb_props);

    const api_lambda_props: ApiGatewayToLambdaProps = {
      lambdaFunctionProps: {
        code: lambda.Code.fromAsset('lambda'),
        runtime: lambda.Runtime.NODEJS_12_X,
        handler: 'hello.handler'
      },
      apiGatewayProps: {
        defaultMethodOptions: {
          authorizationType: api.AuthorizationType.NONE
        }
      }
    };
```

```
    new ApiGatewayToLambda(this, 'ApiGatewayToLambda', api_lambda_props);
  }
}
```

Python

Editar el archivo`hello_constructs/hello_constructs_stack.py`con lo siguiente:

```
from aws_cdk import (
    aws_lambda as _lambda,
    aws_apigateway as apigw,
    aws_dynamodb as ddb,
    core,
)

from aws_solutions_constructs import (
    aws_apigateway_lambda as apigw_lambda,
    aws_lambda_dynamodb as lambda_ddb
)

class HelloConstructsStack(core.Stack):

    def __init__(self, scope: core.Construct, id: str, **kwargs) -> None:
        super().__init__(scope, id, **kwargs)

        # The code that defines your stack goes here

        self.hello_func = _lambda.Function(
          self, 'HelloHandler',
          runtime=_lambda.Runtime.PYTHON_3_7,
          handler='hello.handler',
          code=_lambda.Code.asset('lambda'),
        )

        #  hit counter, aws-lambda-dynamodb pattern
        self.hit_counter = lambda_ddb.LambdaToDynamoDB(
            self, 'LambdaToDynamoDB',
            lambda_function_props=_lambda.FunctionProps(
                runtime=_lambda.Runtime.PYTHON_3_7,
                code=_lambda.Code.asset('lambda'),
                handler='hitcounter.handler',
```

```
                environment={
                    'DOWNSTREAM_FUNCTION_NAME': self.hello_func.function_name
                }
            ),
            dynamo_table_props=ddb.TableProps(
                table_name='Hits',
                partition_key={
                    'name': 'path',
                    'type': ddb.AttributeType.STRING
                }
            )
        )

        apigw_lambda.ApiGatewayToLambda(
            self, 'ApiGatewayToLambda',
            lambda_function_props=_lambda.FunctionProps(
                runtime=_lambda.Runtime.PYTHON_3_7,
                code=_lambda.Code.asset('lambda'),
                handler='hello.handler',
            ),
            api_gateway_props=apigw.RestApiProps(
                default_method_options=apigw.MethodOptions(
                    authorization_type=apigw.AuthorizationType.NONE
                )
            )
        )
```

A continuación, tenemos que conceder la función Contador de visitas creada a partir de la`aws-lambda-dynamodb`patrón agregado anteriormente permiso para invocar nuestra función Hello.

TypeScript

Editar el archivo`lib/hello-constructs.ts`con lo siguiente:

```
  import * as cdk from '@aws-cdk/core';
import * as lambda from '@aws-cdk/aws-lambda';
import * as api from '@aws-cdk/aws-apigateway';
import * as dynamodb from '@aws-cdk/aws-dynamodb';
import { ApiGatewayToLambda, ApiGatewayToLambdaProps } from '@aws-solutions-
constructs/aws-apigateway-lambda';
```

```
import { LambdaToDynamoDB, LambdaToDynamoDBProps } from '@aws-solutions-constructs/
aws-lambda-dynamodb';

export class HelloConstructsStack extends cdk.Stack {
  constructor(scope: cdk.Construct, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    // The code that defines your stack goes here

    // hello function responding to http requests
    const helloFunc = new lambda.Function(this, 'HelloHandler', {
      runtime: lambda.Runtime.NODEJS_12_X,
      code: lambda.Code.fromAsset('lambda'),
      handler: 'hello.handler'
    });

    // hit counter, aws-lambda-dynamodb pattern
    const lambda_ddb_props: LambdaToDynamoDBProps = {
      lambdaFunctionProps: {
          code: lambda.Code.asset(`lambda`),
          runtime: lambda.Runtime.NODEJS_12_X,
          handler: 'hitcounter.handler',
          environment: {
              DOWNSTREAM_FUNCTION_NAME: helloFunc.functionName
          }
      },
      dynamoTableProps: {
          tableName: 'Hits',
          partitionKey: { name: 'path', type: dynamodb.AttributeType.STRING }
      }
    };

    const hitcounter = new LambdaToDynamoDB(this, 'LambdaToDynamoDB',
 lambda_ddb_props);

    // grant the hitcounter lambda role invoke permissions to the hello function
    helloFunc.grantInvoke(hitcounter.lambdaFunction);

    const api_lambda_props: ApiGatewayToLambdaProps = {
      lambdaFunctionProps: {
        code: lambda.Code.fromAsset('lambda'),
        runtime: lambda.Runtime.NODEJS_12_X,
        handler: 'hello.handler'
      },
```

```
      apiGatewayProps: {
        defaultMethodOptions: {
          authorizationType: api.AuthorizationType.NONE
        }
      }
    };

    new ApiGatewayToLambda(this, 'ApiGatewayToLambda', api_lambda_props);
  }
}
```

Python

Editar el archivo`hello_constructs/hello_constructs_stack.py`con lo siguiente:

```
from aws_cdk import (
    aws_lambda as _lambda,
    aws_apigateway as apigw,
    aws_dynamodb as ddb,
    core,
)

from aws_solutions_constructs import (
    aws_apigateway_lambda as apigw_lambda,
    aws_lambda_dynamodb as lambda_ddb
)

class HelloConstructsStack(core.Stack):

    def __init__(self, scope: core.Construct, id: str, **kwargs) -> None:
        super().__init__(scope, id, **kwargs)

        # The code that defines your stack goes here

        self.hello_func = _lambda.Function(
          self, 'HelloHandler',
          runtime=_lambda.Runtime.PYTHON_3_7,
          handler='hello.handler',
          code=_lambda.Code.asset('lambda'),
        )

        #  hit counter, aws-lambda-dynamodb pattern
```

```
        self.hit_counter = lambda_ddb.LambdaToDynamoDB(
            self, 'LambdaToDynamoDB',
            lambda_function_props=_lambda.FunctionProps(
                runtime=_lambda.Runtime.PYTHON_3_7,
                code=_lambda.Code.asset('lambda'),
                handler='hitcounter.handler',
                environment={
                    'DOWNSTREAM_FUNCTION_NAME': self.hello_func.function_name
                }
            ),
            dynamo_table_props=ddb.TableProps(
                table_name='Hits',
                partition_key={
                    'name': 'path',
                    'type': ddb.AttributeType.STRING
                }
            )
        )

        # grant the hitcounter lambda role invoke permissions to the hello function
        self.hello_func.grant_invoke(self.hit_counter.lambda_function)

        apigw_lambda.ApiGatewayToLambda(
            self, 'ApiGatewayToLambda',
            lambda_function_props=_lambda.FunctionProps(
                runtime=_lambda.Runtime.PYTHON_3_7,
                code=_lambda.Code.asset('lambda'),
                handler='hello.handler',
            ),
            api_gateway_props=apigw.RestApiProps(
                default_method_options=apigw.MethodOptions(
                    authorization_type=apigw.AuthorizationType.NONE
                )
            )
        )
```

Por último, tenemos que actualizar nuestro`aws-apigateway-lambda`para utilizar nuestra nueva función de contador de visitas que se aprovisionó con el`aws-lambda-dynamodb`patrón anterior.

TypeScript

Editar el archivo`lib/hello-constructs.ts`con lo siguiente:

```
  import * as cdk from '@aws-cdk/core';
import * as lambda from '@aws-cdk/aws-lambda';
import * as api from '@aws-cdk/aws-apigateway';
import * as dynamodb from '@aws-cdk/aws-dynamodb';
import { ApiGatewayToLambda, ApiGatewayToLambdaProps } from '@aws-solutions-
constructs/aws-apigateway-lambda';
import { LambdaToDynamoDB, LambdaToDynamoDBProps } from '@aws-solutions-constructs/
aws-lambda-dynamodb';

export class HelloConstructsStack extends cdk.Stack {
  constructor(scope: cdk.Construct, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    // The code that defines your stack goes here

    // hello function responding to http requests
    const helloFunc = new lambda.Function(this, 'HelloHandler', {
      runtime: lambda.Runtime.NODEJS_12_X,
      code: lambda.Code.fromAsset('lambda'),
      handler: 'hello.handler'
    });

    // hit counter, aws-lambda-dynamodb pattern
    const lambda_ddb_props: LambdaToDynamoDBProps = {
      lambdaFunctionProps: {
          code: lambda.Code.asset(`lambda`),
          runtime: lambda.Runtime.NODEJS_12_X,
          handler: 'hitcounter.handler',
          environment: {
              DOWNSTREAM_FUNCTION_NAME: helloFunc.functionName
          }
      },
      dynamoTableProps: {
          tableName: 'Hits',
          partitionKey: { name: 'path', type: dynamodb.AttributeType.STRING }
      }
    };
```

```
    const hitcounter = new LambdaToDynamoDB(this, 'LambdaToDynamoDB',
 lambda_ddb_props);

    // grant the hitcounter lambda role invoke permissions to the hello function
    helloFunc.grantInvoke(hitcounter.lambdaFunction);

    const api_lambda_props: ApiGatewayToLambdaProps = {
      existingLambdaObj: hitcounter.lambdaFunction,
      apiGatewayProps: {
        defaultMethodOptions: {
          authorizationType: api.AuthorizationType.NONE
        }
      }
    };

    new ApiGatewayToLambda(this, 'ApiGatewayToLambda', api_lambda_props);
  }
}
```

Python

Editar el archivo`hello_constructs/hello_constructs_stack.py`con lo siguiente:

```
from aws_cdk import (
    aws_lambda as _lambda,
    aws_apigateway as apigw,
    aws_dynamodb as ddb,
    core,
)

from aws_solutions_constructs import (
    aws_apigateway_lambda as apigw_lambda,
    aws_lambda_dynamodb as lambda_ddb
)

class HelloConstructsStack(core.Stack):

    def __init__(self, scope: core.Construct, id: str, **kwargs) -> None:
        super().__init__(scope, id, **kwargs)

        # The code that defines your stack goes here
```

```
        self.hello_func = _lambda.Function(
          self, 'HelloHandler',
          runtime=_lambda.Runtime.PYTHON_3_7,
          handler='hello.handler',
          code=_lambda.Code.asset('lambda'),
        )

        #  hit counter, aws-lambda-dynamodb pattern
        self.hit_counter = lambda_ddb.LambdaToDynamoDB(
            self, 'LambdaToDynamoDB',
            lambda_function_props=_lambda.FunctionProps(
                runtime=_lambda.Runtime.PYTHON_3_7,
                code=_lambda.Code.asset('lambda'),
                handler='hitcounter.handler',
                environment={
                    'DOWNSTREAM_FUNCTION_NAME': self.hello_func.function_name
                }
            ),
            dynamo_table_props=ddb.TableProps(
                table_name='Hits',
                partition_key={
                    'name': 'path',
                    'type': ddb.AttributeType.STRING
                }
            )
        )

        # grant the hitcounter lambda role invoke permissions to the hello function
        self.hello_func.grant_invoke(self.hit_counter.lambda_function)

        apigw_lambda.ApiGatewayToLambda(
            self, 'ApiGatewayToLambda',
            existing_lambda_obj=self.hit_counter.lambda_function,
            api_gateway_props=apigw.RestApiProps(
                default_method_options=apigw.MethodOptions(
                    authorization_type=apigw.AuthorizationType.NONE
                )
            )
        )
```

Revisar los cambios

Construyamos nuestro proyecto y revisemos los cambios en nuestros recursos que se producirán cuando implementemos esto:

```
npm run build
cdk diff
```

Nuestra salida debe tener el siguiente aspecto:

```
Stack HelloConstructsStack
IAM Statement Changes
#######################################################################################
#   # Resource                          # Effect # Action                          #
 Principal                          # Condition #
#######################################################################################
# + # ${HelloHandler.Arn}               # Allow  # lambda:InvokeFunction            #
 AWS:${LambdaFunctionServiceRole}   #           #
#######################################################################################
# + # ${HelloHandler/ServiceRole.Arn}   # Allow  # sts:AssumeRole                   #
 Service:lambda.amazonaws.com       #           #
#######################################################################################
# + # ${LambdaToDynamoDB/DynamoTable.Ar # Allow  # dynamodb:BatchGetItem            #
 AWS:${LambdaFunctionServiceRole}   #           #
#   # n}                                #        # dynamodb:BatchWriteItem          #
                                    #           #
#   #                                 #        # dynamodb:DeleteItem              #
                                    #           #
#   #                                 #        # dynamodb:GetItem                 #
                                    #           #
#   #                                 #        # dynamodb:GetRecords              #
                                    #           #
#   #                                 #        # dynamodb:GetShardIterator        #
                                    #           #
#   #                                 #        # dynamodb:PutItem                 #
                                    #           #
#   #                                 #        # dynamodb:Query                   #
                                    #           #
#   #                                 #        # dynamodb:Scan                    #
                                    #           #
```

```
#   #                                   #        # dynamodb:UpdateItem              #
                                    #           #
#######################################################################################
IAM Policy Changes
#######################################################################################
#   # Resource                     # Managed Policy ARN
                             #
#######################################################################################
# + # ${HelloHandler/ServiceRole} # arn:${AWS::Partition}:iam::aws:policy/service-role/
AWSLambdaBasicExecutionRole #
#######################################################################################
(NOTE: There may be security-related changes not in this list. See https://github.com/
aws/aws-cdk/issues/1299)

Resources
[+] AWS::IAM::Role HelloHandler/ServiceRole HelloHandlerServiceRole11EF7C63
[+] AWS::Lambda::Function HelloHandler HelloHandler2E4FBA4D
[+] AWS::DynamoDB::Table LambdaToDynamoDB/DynamoTable
 LambdaToDynamoDBDynamoTable53C1442D
[+] AWS::IAM::Policy LambdaFunctionServiceRole/DefaultPolicy
 LambdaFunctionServiceRoleDefaultPolicy126C8897
[~] AWS::Lambda::Function LambdaFunction LambdaFunctionBF21E41F
 ## [+] Environment
 #   ## {"Variables":{"DOWNSTREAM_FUNCTION_NAME":
{"Ref":"HelloHandler2E4FBA4D"},"DDB_TABLE_NAME":
{"Ref":"LambdaToDynamoDBDynamoTable53C1442D"}}}
 ## [~] Handler
 #   ## [-] hello.handler
 #   ## [+] hitcounter.handler
 ## [~] DependsOn
     ## @@ -1,3 +1,4 @@
        [ ] [
        [+]   "LambdaFunctionServiceRoleDefaultPolicy126C8897",
        [ ]   "LambdaFunctionServiceRole0C4CDE0B"
        [ ] ]
```

cdk/

Bien, ¿listo para la implementación?

```
cdk deploy
```

Salida de la pila

Cuando se haya completado la implementación, notará lo siguiente:

```
Outputs:
HelloConstructsStack.RestApiEndpoint0551178A = https://xxxxxxxxxx.execute-api.us-
east-1.amazonaws.com/prod/
```

Probar su aplicación

Tratemos de golpear este punto final con rizo. Copie la URL y ejecute (es probable que su prefijo y región sean diferentes).

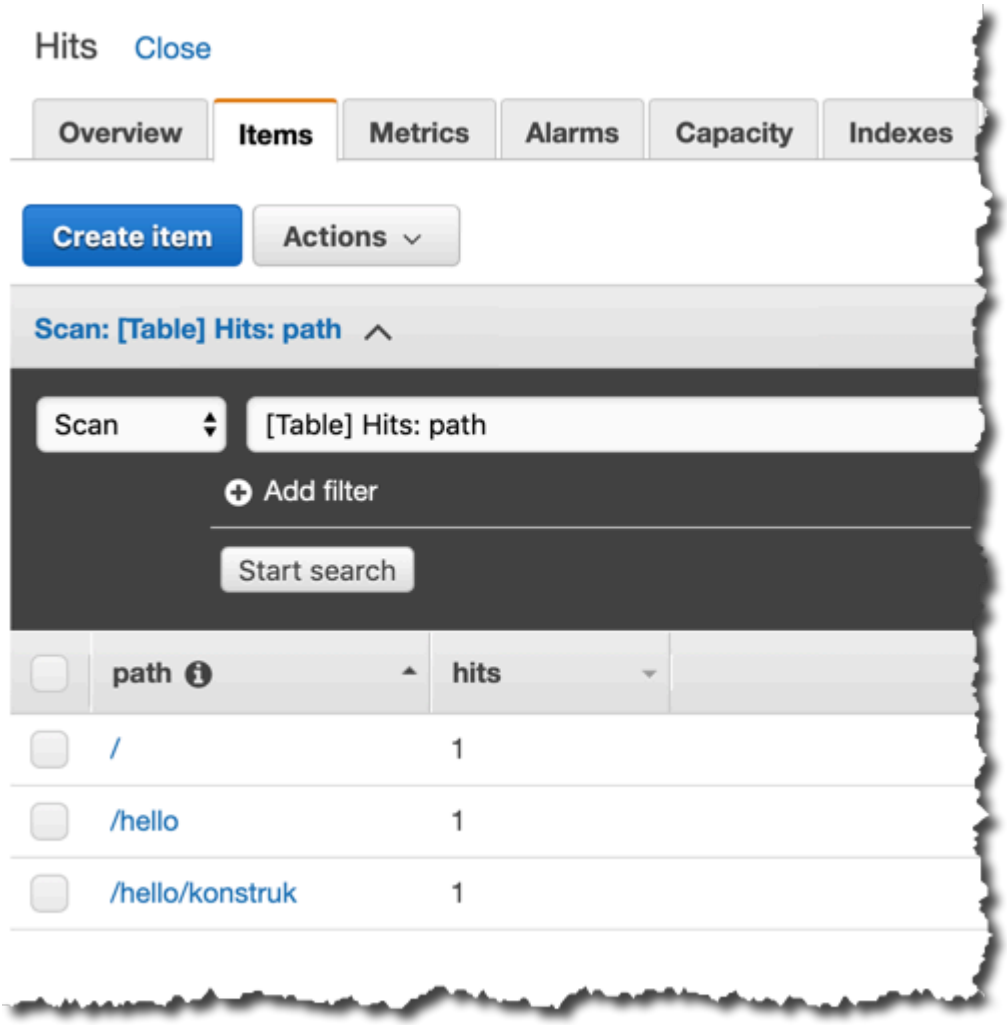
```
curl https://xxxxxxxxxx.execute-api.us-east-1.amazonaws.com/prod/
```

La salida debe tener el siguiente aspecto:

```
Hello, AWS Solutions Constructs! You've hit /
```

Ahora vamos a revisar el`Hits`Tabla de Amazon DynamoDB.

1. Vaya a la consola de DynamoDB.
2. Asegúrese de que se encuentra en la región donde creó la tabla.
3. SelectTablasen el panel de navegación y seleccione la casillaHitsTabla de.
4. Abra la tabla y seleccione «Elementos».
5. Deberías ver cuántos aciertos tienes por cada camino.



6. Intente acceder a una nueva ruta y actualice la vista Elementos. Debería ver un nuevo elemento con un`hits`cuenta de uno.

Si esta es la salida que recibiste, ¡tu aplicación funciona!

Casos de uso de ejemplo

Esta biblioteca incluye una colección de implementaciones de casos de uso funcionales para demostrar el uso de patrones arquitectónicos de componentes fijos. Estos se pueden usar de la misma manera que los patrones arquitectónicos, y pueden ser conceptualizados como una abstracción adicional de «nivel superior» de esos patrones. Los siguientes casos de uso se proporcionan como ejemplos funcionales:

Sitio web estático de AWS S3

Este patrón de caso de uso (`aws-s3-static-website`) implementa una distribución de Amazon CloudFront, un depósito de Amazon S3 y un recurso personalizado basado en AWS Lambda para copiar el contenido del sitio web estático para el sitio web de demostración de Wild Rydes (parte de`aws-serverless-web-app`Aplicación).

> Código fuente (aws-s3-static-website)
>
> https://github.com/awslabs/aws-solutions-constructs/tree/master/source/use_cases/aws-s3-static-website

Controlador de imágenes sencillo sin servidor de AWS

Este patrón de caso de uso (`aws-serverless-image-handler`) implementa una distribución de Amazon CloudFront, una API de REST de Amazon API Gateway, una función de AWS Lambda y los permisos/lógicos necesarios para aprovisionar una API de controlador de imágenes funcional para servir contenido de imagen desde uno o más depósitos de Amazon S3 dentro de la cuenta de implementación.

> Código fuente (aws-serverless-image-handler)
>
> https://github.com/awslabs/aws-solutions-constructs/tree/master/source/use_cases/aws-serverless-image-handler

Aplicación web sin servidor de AWS

Este patrón de caso de uso (`aws-serverless-web-app`) implementa una sencilla aplicación web sin servidor que permite a los usuarios solicitar viajes en unicornio a la flota de Wild Rydes. La aplicación presentará a los usuarios una interfaz de usuario basada en HTML para indicar la ubicación donde les gustaría ser recogidos y se interconectará en el back-end con un servicio web RESTful para enviar la solicitud y enviar un unicornio cercano. La aplicación también proporcionará facilidades para que los usuarios se registren en el servicio e inicien sesión antes de solicitar viajes.

Código fuente (aws-serverless-web-app)

https://github.com/awslabs/aws-solutions-constructs/tree/master/source/use_cases/aws-serverless-web-app

Referencia de la API

AWS Solutions Constructs (Constructs) es una extensión de código abierto del AWS Cloud Development Kit (AWS CDK) que proporciona patrones multiservicio y bien diseñados para definir rápidamente soluciones en código con el fin de crear una infraestructura predecible y repetible. El objetivo de Constructs es acelerar la experiencia de los desarrolladores para crear soluciones de cualquier tamaño utilizando definiciones basadas en patrones para su arquitectura.

Los patrones definidos en los componentes fijos son abstracciones multiservicio de alto nivel de construcciones CDK de AWS que tienen configuraciones predeterminadas basadas en prácticas recomendadas bien diseñadas. La biblioteca está organizada en módulos lógicos utilizando técnicas orientadas a objetos para crear cada modelo de patrón arquitectónico.

El CDK está disponible en los siguientes lenguajes:

- JavaScript, TypeScript (Node.js ≥ 10.3.0)
- Python (Python ≥ 3.6)
- Java (Java ≥ 1.8)

Modules

AWS Solutions Constructs se organiza en varios módulos. Se llaman así:

- aws-xxx: Paquete de patrones bien diseñado para los servicios indicados. Este paquete contendrá construcciones que contienen varios módulos de servicio AWS CDK para configurar el patrón dado.
- xxx: Paquetes que no empiezan»Aws-«son módulos centrales de construcción que se utilizan para configurar los valores predeterminados de las mejores prácticas para los servicios utilizados dentro de la biblioteca de patrones.

Contenido Module

Los módulos contienen los siguientes tipos:

- Patrones- Todas las construcciones multiservicios de nivel superior en esta biblioteca.
- Otros Tipos- Todas las clases, interfaces, estructuras y enumeraciones no constructivas que existen para soportar los patrones.

Los patrones toman un conjunto de propiedades (de entrada) en su constructor; el conjunto de propiedades (y cuáles son requeridas) se puede ver en la página de documentación de un patrón.

La página de documentación del patrón también enumera los métodos disponibles para llamar y las propiedades que se pueden utilizar para recuperar información sobre el patrón después de que se haya creado una instancia.

aws-apigateway-dynamodb

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_apigateway_dynamodb`
TypeScript	`@aws-solutions-constructs/aws-apigateway-dynamodb`
Java	`software.amazon.awsconstructs.services.apigatewaydynamodb`

Overview

Este constructo de soluciones de AWS implementa una API REST de Amazon API Gateway conectada a una tabla de Amazon DynamoDB.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { ApiGatewayToDynamoDBProps, ApiGatewayToDynamoDB } from "@aws-solutions-
constructs/aws-apigateway-dynamodb";

new ApiGatewayToDynamoDB(this, 'test-api-gateway-dynamodb-default', {});
```

Initializer

```
new ApiGatewayToDynamoDB(scope: Construct, id: string, props:
 ApiGatewayToDynamoDBProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`ApiGatewayToDynamoDBProps`

Patrón de construcción

Nombre	Tipo	Descripción
DynamoTableProps	`dynamodb.TableProps`	Los accesorios proporcio nados por el usuario opcionale s para anular los accesorios predeterminados para la tabla de DynamoDB
¿ApigatewayProps?	`api.RestApiProps`	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para API Gateway.
AllowCreateOperation	`boolean`	Si desea implementar el método de puerta de API

Nombre	Tipo	Descripción
		Gateway para la operación Crear en la tabla DynamoDB.
CreateRequestTemplate	`string`	Plantilla de solicitud de API Gateway para el método Create, necesaria si AllowCreateOperation se establece
AllowReadOperation	`boolean`	Si se debe implementar el método API Gateway para la operación de lectura en la tabla DynamoDB.
AllowUpdateOperation	`boolean`	Si desea implementar el método de API Gateway para la operación de actualización en la tabla DynamoDB.
UpdateRequestTemplate	`string`	Plantilla de solicitud de API Gateway para el método Update, necesaria si AllowUpdateOperation se establece en true
AllowDeleteOperation	`boolean`	Si desea implementar el método de puerta de API Gateway para la operación de eliminación en la tabla DynamoDB

Nombre	Tipo	Descripción
LogGroupProps?	`logs.LogGroupProps`	Los accesorios opcionales proporcionados por el usuario para anular los accesorios predeterminados para el grupo de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
ApiGateway	`api.RestApi`	Devuelve una instancia de la API REST de API Gateway creada por el patrón.
ApigatewayCloudWatchRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón que permite el registro de acceso desde API Gateway REST API a CloudWatch.
ApigatewayLogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por el patrón al que se envían los registros de acceso de API REST de API de API Gateway de API.
ApigateWayRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón para la API Gateway.

Nombre	Tipo	Descripción
DynamoTable	`dynamodb.Table`	Devuelve una instancia de la tabla DynamoDB creada por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

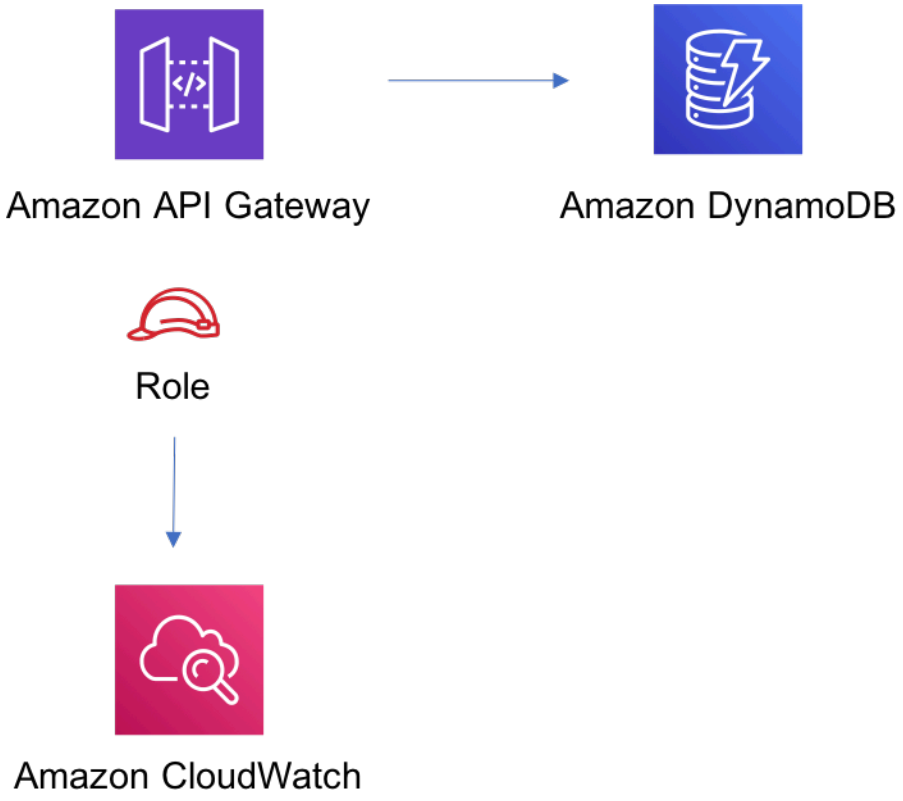
Amazon API Gateway

- Implementación de un extremo de API optimizado para bordes
- Habilitar el registro de CloudWatch para API Gateway
- Configurar el rol de IAM de acceso mínimo con privilegios para API Gateway
- Establezca el AuthorizationType predeterminado para todos los métodos API en IAM
- Habilitar el X-Ray streo

Tabla de Amazon DynamoDB

- Establezca el modo de facturación para la tabla de DynamoDB en Bajo demanda (pago por solicitud)
- Habilitar el cifrado del lado del servidor para DynamoDB Table mediante la clave KMS administrada por AWS
- Crea una clave de partición llamada 'id' para la tabla de DynamoDB
- Conservar la tabla al eliminar la pila CloudFormation
- Activación de copias de seguridad continuas y recuperación a un momento dado

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-apigateway-dynamodb

aws-apigateway-iot

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_apigateway_iot`
Mecanografía	`@aws-solutions-constructs/aws-apigateway-iot`
Java	`software.amazon.awsconstructs.services.apigatewayiot`

Overview

Este constructo de soluciones de AWS implementa una API REST de Amazon API Gateway conectada al patrón AWS IoT.

Esta construcción crea un proxy HTTPS escalable entre API Gateway y AWS IoT. Esto resulta útil cuando se desea permitir que los dispositivos heredados que no admiten el protocolo MQTT o MQTT/WebSocket interactúen con la plataforma AWS IoT.

Esta implementación permite publicar mensajes de sólo escritura en determinados temas de MQTT, y también admite actualizaciones de instantáneas de dispositivos HTTPS para las cosas permitidas en el registro de dispositivos. No implica funciones de Lambda para enviar mensajes proxy, sino que se basa en la integración directa de API Gateway a AWS IoT, que admite tanto mensajes JSON como mensajes binarios.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { ApiGatewayToIot } from '@aws-solutions-constructs/aws-apigateway-iot';

new ApiGatewayToIot(this, 'ApiGatewayToIotPattern', {
    iotEndpoint: 'a1234567890123-ats'
```

```
});
```

Initializer

```
new ApiGatewayToIot(scope: Construct, id: string, props: ApiGatewayToIotProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`ApiGatewayToIotProps`

Patrón de construcción

Nombre	Tipo	Descripción
IOtendPoint	`string`	El subdominio de punto final de AWS IoT con el que integrar API Gateway (por ejemplo, a1234567890123-ats).
ApigateWayCreateApiKey?	`boolean`	Si toma el valor`true`, se crea una clave API y se asocia a un UsagePlan. El usuario debe especificar el encabezado `x-api-key` mientras accede a RestApi. Valor predeterminado establecido en`false`.
ApigatewayExecutionRole?	`iam.Role`	El rol de IAM utilizado por API Gateway para acceder a AWS IoT. Si no se especifica, se crea un rol predeterminado

Nombre	Tipo	Descripción
		con acceso comodín ('*') a todos los temas y elementos.
¿ApigateWayProps?	`api.restApiProps`	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para la API REST de API Gateway.
LogGroupProps?	`logs.LogGroupProps`	Compatibilidad con accesorio s opcionales proporcionados por el usuario para anular los accesorios predeterminados para el grupo de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
ApiGateway	`api.RestApi`	Devuelve una instancia de la API REST de API Gateway creada por el patrón.
ApigatewayCloudWatchRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón que permite el registro de acceso desde API Gateway REST API a CloudWatch.
ApigatewayLogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por el patrón al que se envían los registros de acceso de API

Nombre	Tipo	Descripción
		REST de API de API Gateway de API.
ApigateWayRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón para la API Gateway.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Amazon API Gateway

- Implementación de un extremo de API optimizado para bordes
- Crea recursos de API con`POST`Método para publicar mensajes en temas de IoT Temas
- Crea recursos de API con`POST`Método para publicar mensajes en`ThingShadow`y`NamedShadows`
- Permita que el registro de CloudWatch para
- Configurar el rol de IAM para API Gateway con acceso a todos los temas y cosas
- Establezca el AuthorizationType predeterminado para todos los métodos API en IAM
- Permita X-Ray Rastreo
- Crea un UsagePlan y se asocia a`prod`stage

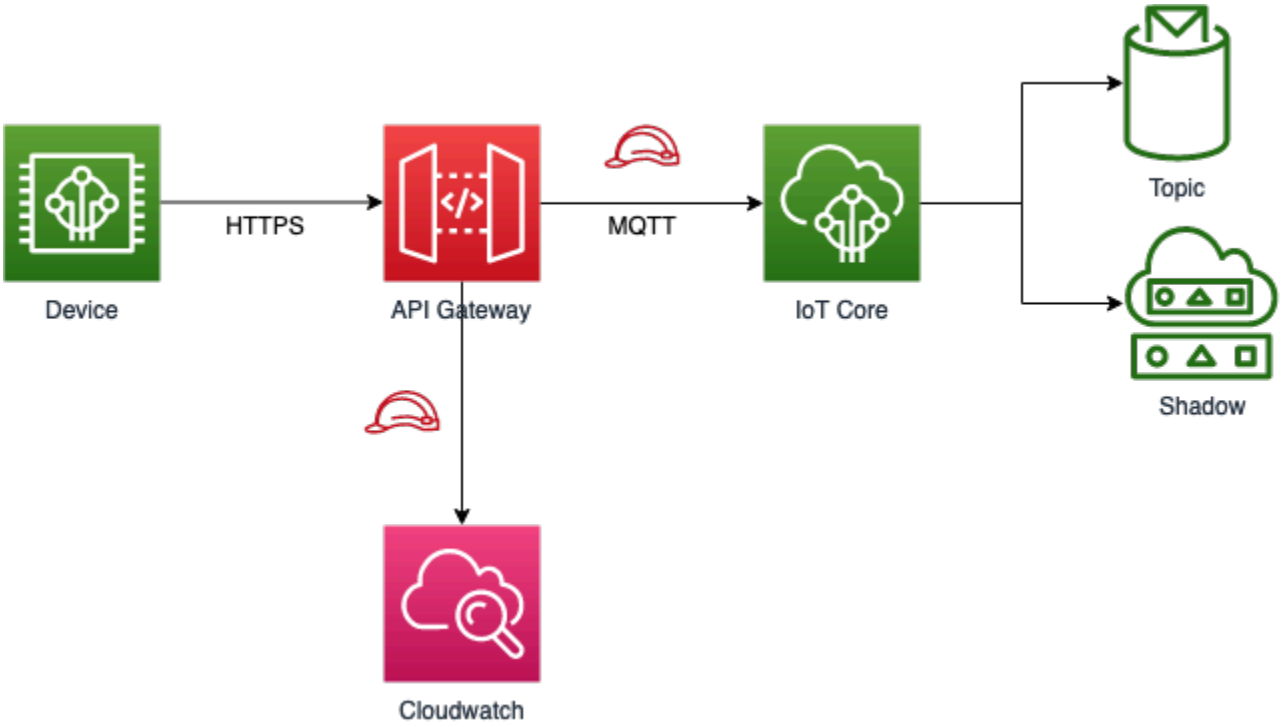
A continuación se muestra una descripción de los diferentes recursos y métodos expuestos por la API Gateway después de implementar el constructo. Consulte laEjemplospara obtener más información sobre cómo probar fácilmente estos extremos usando`curl`.

Método	Recurso	Parámetros de consulta	Codigos devueltos	Descripción
`POST`	`/message/<topics>`	`qos`	`200/403/500`	Al llamar a este punto final, debe pasar los temas en los que desea publicar

Método	Recurso	Parámetros de consulta	Codigos devueltos	Descripción
				(por ejemplo,`/message/device/foo` `).
POST	`/shadow/<thingName>`	Ninguno	`200/403/500`	Esta ruta permite actualizar el documento sombra de una cosa, dado `suthingName` con el tipo de sombra sin nombre (clásico). El cuerpo deberá cumplir la estutura normalizada de sombras que comprende `unstate`nodo y asociado`desiredyrepor` Consulte la[Actualización de la sombra del dispositivo](#)para ver un ejemplo.

Método	Recurso	Parámetros de consulta	Codigos devueltos	Descripción
POST	/shadow/<thingName>/<shadowName>	Ninguno	200/403/500	Esta ruta permite actualizar el documento de sombra nombrado de una cosa, dado suthingName y lashadowName utilizando el tipo de sombra con nombre asignado. El cuerpo deberá cumplir la estutura normalizada de sombras que comprende unstatenodo y asociadodesiredyrepor Consulte laActualización de las sombras guardadaspara ver un ejemplo.

Architecture



Examples

Los siguientes ejemplos solo funcionan con`API_KEY`, ya que la autorización de IAM requiere que también se especifique un token SIGv4, asegúrese de que la propiedad`apiGatewayCreateApiKey`de los accesorios de construcción se establece `entrue`mientras despliega la pila, de lo contrario, los siguientes ejemplos no funcionarán.

Publicación de un mensaje

Puede usar`curl`para publicar un mensaje en diferentes temas de MQTT utilizando la API HTTPS. El siguiente ejemplo publicará un mensaje en el`device/foo`Tema.

```
curl -XPOST https://<stage-id>.execute-api.<region>.amazonaws.com/prod/message/device/
foo -H "x-api-key: <api-key>" -H "Content-Type: application/json" -d '{"Hello":
 "World"}'
```

Nota: Reemplace la`stage-id`,`region`, y`api-key`con los valores de implementación.

Puede encadenar nombres de temas en la URL y la API acepta hasta 7 subtemas en los que puede publicar. Por ejemplo, en el siguiente ejemplo publica un mensaje en el tema.`device/foo/bar/abc/xyz`.

```
curl -XPOST https://<stage-id>.execute-api.<region>.amazonaws.com/prod/message/device/
foo/bar/abc/xyz -H "x-api-key: <api-key>" -H "Content-Type: application/json" -d
 '{"Hello": "World"}'
```

Actualización de la sombra del dispositivo

Para actualizar el documento de sombra asociado a una cosa determinada, puede emitir una solicitud de estado de sombra usando un nombre de cosa. Consulte el siguiente ejemplo de cómo actualizar una sombra de objeto.

```
curl -XPOST https://<stage-id>.execute-api.<region>.amazonaws.com/prod/shadow/device1 -
H "x-api-key: <api-key>" -H "Content-Type: application/json" -d '{"state": {"desired":
 { "Hello": "World" }}}'
```

Actualización de las sombras guardadas

Para actualizar el documento de sombra asociado a la sombra con nombre de una cosa determinada, puede emitir una solicitud de estado de sombra utilizando un nombre de cosa y un nombre de sombra. Vea el siguiente ejemplo sobre cómo actualizar una sombra con nombre.

```
curl -XPOST https://<stage-id>.execute-api.<region>.amazonaws.com/prod/shadow/device1/
shadow1 -H "x-api-key: <api-key>" -H "Content-Type: application/json" -d '{"state":
 {"desired": { "Hello": "World" }}}'
```

Envío de cargas binarias

Es posible enviar una carga útil binaria a la API proxy, hasta el servicio AWS IoT. En el siguiente ejemplo, enviamos el contenido de la`README.md`asociado con este módulo (tratado como datos binarios) a`device/foo`con el tema de`application/octet-stream`Tipo de contenido.

```
curl -XPOST https://<stage-id>.execute-api.<region>.amazonaws.com/prod/message/device/
foo/bar/baz/qux -H "x-api-key: <api-key>" -H "Content-Type: application/octet-stream"
 --data-binary @README.md
```

Nota: Ejecute este comando mientras esté en el directorio de este proyecto. A continuación, puede probar el envío de otro tipo de archivos binarios desde su sistema de archivos.

GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-apigateway-iot

aws-apigateway-kinesisstreams

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersionado semánticoEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_apigateway_kinesisstreams`
Mecanografía	`@aws-solutions-constructs/aws-apigateway-kinesisstreams`

Idioma	Paquete
Java	`software.amazon.awsconstructs.services.apigatewaykinesisstreams`

Overview

Este patrón implementa una API REST de Amazon API Gateway conectada a un flujo de datos de Amazon Kinesis.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { ApiGatewayToKinesisStreams, ApiGatewayToKinesisStreamsProps } from '@aws-solutions-constructs/aws-apigateway-kinesisstreams';

new ApiGatewayToKinesisStreams(this, 'test-apigw-kinesis', {});
```

Initializer

```
new ApiGatewayToKinesisStreams(scope: Construct, id: string, props:
 ApiGatewayToKinesisStreamsProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`ApiGatewayToKinesisStreamsProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ApigateWayProps?	`api.RestApiProps`	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para la API REST de API Gateway.
PutRecordRequestTemplate?	`string`	Plantilla de solicitud de API Gateway para la acción PutRecord. Si no se proporcio na, se usará uno predeterm inado.
PutRecordRequestModel?	`api.ModelOptions`	Modelo de solicitud de API Gateway para la acción PutRecord. Si no se proporcio na, se creará uno predeterm inado.
PutRecordsRequestTemplate?	`string`	Plantilla de solicitud de API Gateway para la acción PutRecords Si no se proporcio na, se usará uno predeterm inado.
PutRecordRequestModel?	`api.ModelOptions`	Modelo de solicitud de API Gateway para la acción PutRecords. Si no se proporciona, se creará uno predeterminado.
¿ExistenteStreamObj?	`kinesis.Stream`	Instancia existente de Kinesis Stream, proporcionando tanto esto como`kinesisSt`

Nombre	Tipo	Descripción
		`reamProps` provocará un error.
¿KinesisStreamProps?	`kinesis.StreamProps`	Props opcionales proporcionados por el usuario para anular los apoyos predeterminados para la transmisión de Kinesis.
LogGroupProps?	`logs.LogGroupProps`	Aplicaciones opcionales proporcionadas por el usuario para anular los accesorios predeterminados para el grupo de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
ApiGateway	`api.RestApi`	Devuelve una instancia de la API REST de API Gateway creada por el patrón.
ApigateWayRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón para la API de REST de puerta de enlace de API.
ApigatewayCloudWatchRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón que permite el registro de acceso desde API Gateway REST API a CloudWatch.

Nombre	Tipo	Descripción
ApigatewayLogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por el patrón al que se envían los registros de acceso de API REST de API de API Gateway de API.
KinesisStream	`kinesis.Stream`	Devuelve una instancia de la secuencia Kinesis creada por el patrón.

Ejemplo de uso de las API

Método	Ruta de solicitud	Cuerpo de la solicitud	Acción de cola	Descripción
POST	`/record`	`{ "data": "Hello World!", "partitionKey": "pk001" }`	`kinesis:PutRecord`	Escribe un solo registro de datos en el flujo.
POST	`/records`	`{ "records": [ { "data": "abc", "partitionKey": "pk001"`	`kinesis:PutRecords`	Escribe varios registros de datos en la secuencia en una sola llamada.

Método	Ruta de solicitud	Cuerpo de la solicitud	Acción de cola	Descripción

		`},`
		
		`{ "data":`
		`"xyz",`
		
		`"partitio`
		`nKey":`
		`"pk001"`
		`}`
		`]`
		`}`		

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

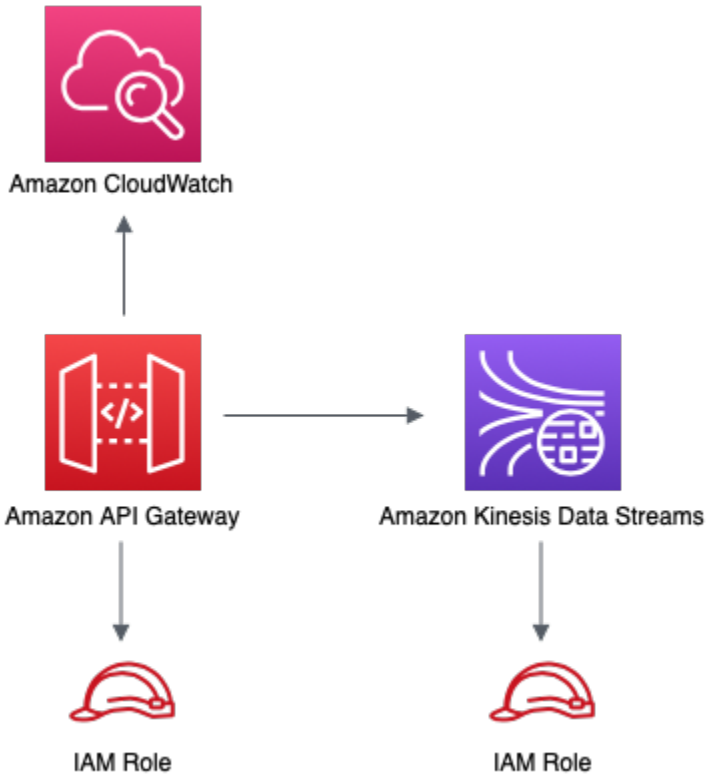
Amazon API Gateway

- Implemente un extremo de la API optimizada para bordes.
- Active el registro de CloudWatch para API Gateway.
- Configurar el rol de IAM de acceso mínimo con privilegios para API Gateway.
- Establezca el AuthorizationType predeterminado para todos los métodos API en IAM.
- Active el rastreo de X-Ray.
- Validar el cuerpo de la solicitud antes de pasar los datos a Kinesis.

Amazon Kinesis Data Stream

- Configure el rol de IAM de acceso mínimo con privilegios para la transmisión de Kinesis.
- Habilite el cifrado del lado del servidor para Kinesis Stream mediante la clave KMS administrada de AWS.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-apigateway-kinesisstreams

aws-apigateway-lambda

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasmodelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_apigateway_lambda`
Mecanografía	`@aws-solutions-constructs/aws-apigateway-lambda`
Java	`software.amazon.awsconstructs.services.apigatewaylambda`

Overview

Este constructo de soluciones de AWS implementa una API REST de Amazon API Gateway conectada a una función de AWS Lambda.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { ApiGatewayToLambda } from '@aws-solutions-constructs/aws-apigateway-lambda';

new ApiGatewayToLambda(this, 'ApiGatewayToLambdaPattern', {
    lambdaFunctionProps: {
      runtime: lambda.Runtime.NODEJS_14_X,
      // This assumes a handler function in lib/lambda/index.js
      code: lambda.Code.fromAsset(`${__dirname}/lambda`),
      handler: 'index.handler'
    }
});
```

Initializer

```
new ApiGatewayToLambda(scope: Construct, id: string, props: ApiGatewayToLambdaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`ApiGatewayToLambdaProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si un`existingLambdaObj` se encuentra.
¿ApigateWayProps?	`api.LambdaRestApiProps`	Props opcionales proporcionados por el usuario para anular los apoyos predeterminados para la API.
LogGroupProps?	`logs.LogGroupProps`	Los accesorios opcionales proporcionados por el usuario para anular los accesorios predeterminados para el grupo

Nombre	Tipo	Descripción
		de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
ApigatewayCloudWatchRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón que permite el registro de acceso desde API Gateway REST API a CloudWatch.
ApigatewayLogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por el patrón al que se envían los registros de acceso de API REST de API de API Gateway de API.
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.
ApiGateway	`api.LambdaRestApi`	Devuelve una instancia de la API REST de API Gateway creada por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

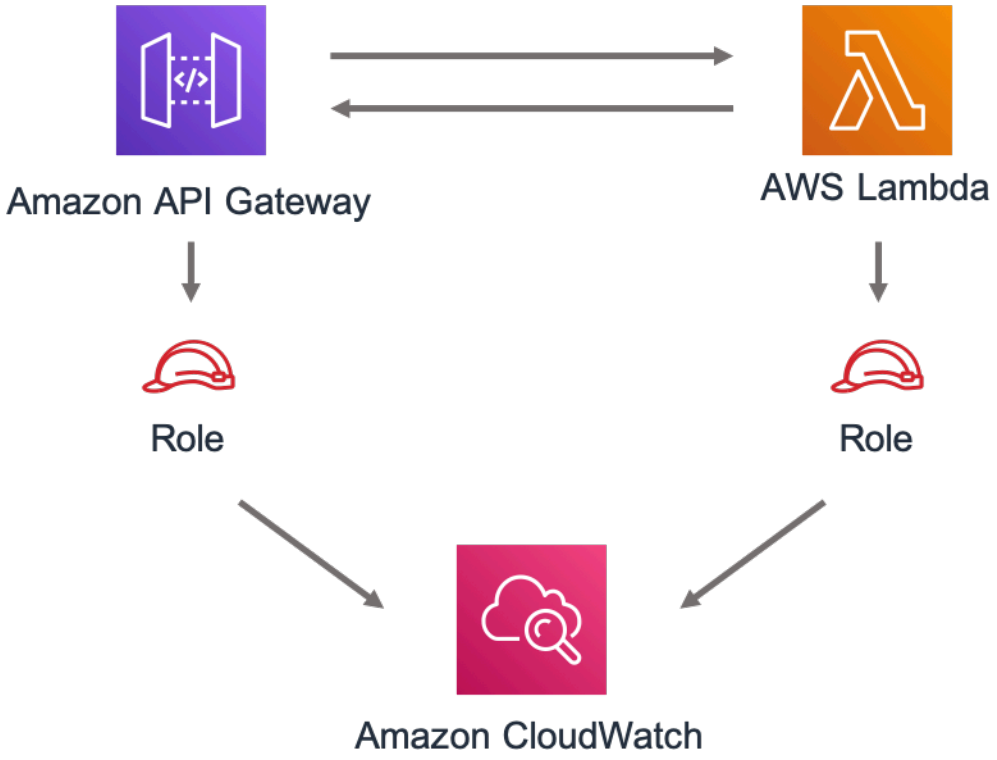
Amazon API Gateway

- Implementación de un extremo de API optimizado para bordes
- Habilitar el registro de CloudWatch para API Gateway
- Configurar el rol de IAM de acceso mínimo con privilegios para API Gateway
- Establezca el AuthorizationType predeterminado para todos los métodos API en IAM
- Habilitación de X-Ray streo
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Función de AWS Lambda

- Configuración de la función de IAM de acceso limitado a privilegios para la función Lambda
- Habilitar conexiones de reutilización con la función Keep-Alive para NodeJS Lambda
- Habilitación de X-Ray streo

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-apigateway-lambda

aws-apigateway-sagemakerendpoint

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Éstos no están sujetos a laVersiones semánticasEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_apigateway_sagemakerendpoint`
Mecanografía	`@aws-solutions-constructs/aws-apigateway-sagemakerendpoint`
Java	`software.amazon.awsconstructs.services.apigatewaysagemakerendpoint`

Overview

Este constructo de soluciones de AWS implementa una API REST de Amazon API Gateway conectada a un extremo de Amazon SageMaker.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { ApiGatewayToSageMakerEndpoint, ApiGatewayToSageMakerEndpointProps } from
 '@aws-solutions-constructs/aws-apigateway-sagemakerendpoint';

// Below is an example VTL (Velocity Template Language) mapping template for mapping
 the Api GET request to the Sagemaker POST request
const requestTemplate =
`{
    "instances": [
#set( $user_id = $input.params("user_id") )
#set( $items = $input.params("items") )
#foreach( $item in $items.split(",") )
    {"in0": [$user_id], "in1": [$item]}#if( $foreach.hasNext ),#end
    $esc.newline
#end
    ]
}`;

// Replace 'my-endpoint' with your Sagemaker Inference Endpoint
new ApiGatewayToSageMakerEndpoint(this, 'test-apigw-sagemakerendpoint', {
    endpointName: 'my-endpoint',
    resourcePath: '{user_id}',
    requestMappingTemplate: requestTemplate
});
```

Initializer

```
new ApiGatewayToSageMakerEndpoint(scope: Construct, id: string, props:
 ApiGatewayToSageMakerEndpointProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`ApiGatewayToSageMakerEndpointProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ApigateWayProps?	`api.RestApiProps`	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para la API REST de API Gateway.
ApigatewayExecutionRole?	`iam.Role`	Función de IAM utilizada por API Gateway para invocar el punto de enlace de SageMaker. Si no se especifica, se crea un rol predeterminado con acceso a`endpointName` .
EndpointName	`string`	Nombre del extremo de inferencia de SageMaker desplegado.
ResourceName	`string`	Nombre de recurso opcional donde estará disponible el método GET.
resourcePath	`string`	Ruta de acceso del recurso para el método GET. La variable definida aquí puede ser referenciada en`requestMappingTemp late` .
RequestMappingTemplate	`string`	Plantilla de asignación para convertir las solicitudes GET recibidas en la API REST a

Nombre	Tipo	Descripción
		solicitudes POST esperadas por el extremo de SageMaker.
ResponseMappingTemplate	`string`	Plantilla de asignación opcional para convertir las respuestas recibidas desde el extremo de SageMaker.
LogGroupProps?	`logs.LogGroupProps`	Elementos opcionales proporcionados por el usuario para anular los accesorios predeterminados para el grupo de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
ApiGateway	`api.LambdaRestApi`	Devuelve una instancia de la API REST de API Gateway creada por el patrón.
ApigateWayRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón para la API de REST de puerta de enlace de API.
ApigatewayCloudWatchRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón que permite el registro de acceso desde API Gateway REST API a CloudWatch.

Nombre	Tipo	Descripción
ApigatewayLogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por el patrón al que se envían los registros de acceso de API REST de API de API Gateway de API.

Uso de las API de ejemplo

Nota: Cada extremo de SageMaker es único y la respuesta de la API dependerá del modelo implementado. El ejemplo que se da a continuación supone que la muestra deEsta entrada de blog. Para obtener una referencia sobre cómo se implementaría, consulteinteg.apigateway-sagemakerendpoint-overwrite.ts.

Método	Ruta de solicitud	Cadena de consulta	Acción de SageMaker	Descripción
GET	`/321`	`items=101,131,162`	`sagemaker:InvokeEndpoint`	Recupera las predicciones para un usuario y elementos específicos.

Configuración predeterminada

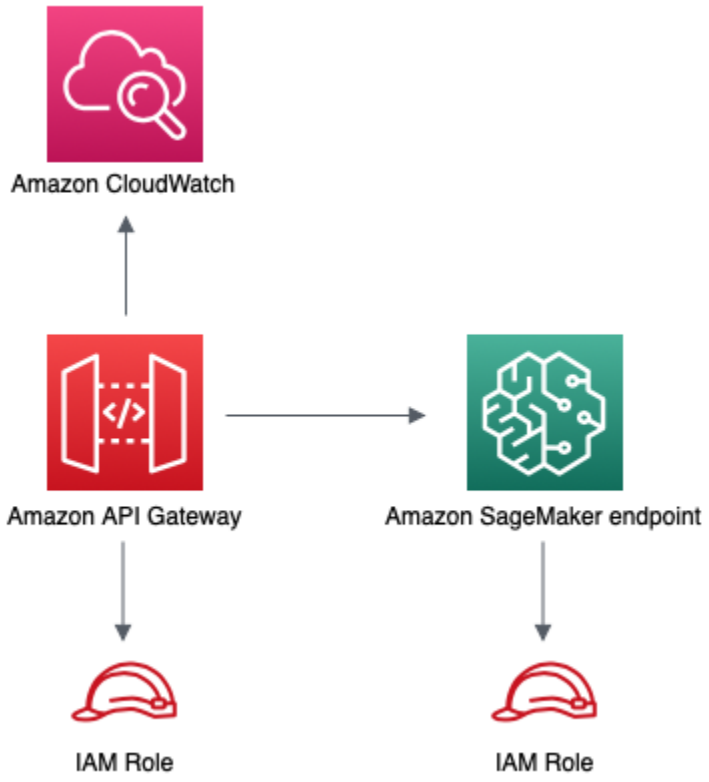
La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Amazon API Gateway

- Implementación de un extremo de API optimizado para bordes
- Habilitar el registro de CloudWatch para API Gateway
- Configurar el rol de IAM de acceso mínimo con privilegios para API Gateway
- Establezca el AuthorizationType predeterminado para todos los métodos API en IAM

- Habilitación de X-Ray streo
- Validar los parámetros de solicitud antes de pasar datos a SageMaker

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-apigateway-sagemakerendpoint

aws-apigateway-sqs

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_apigateway_sqs`
Mecanografiado	`@aws-solutions-constructs/aws-apigateway-sqs`
Java	`software.amazon.awsconstructs.services.apigatewaysqs`

Overview

Este constructo de soluciones de AWS implementa una API REST de Amazon API Gateway conectada a una cola de Amazon SQS.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { ApiGatewayToSqs, ApiGatewayToSqsProps } from "@aws-solutions-constructs/aws-apigateway-sqs";

new ApiGatewayToSqs(this, 'ApiGatewayToSqsPattern', {});
```

Initializer

```
new ApiGatewayToSqs(scope: Construct, id: string, props: ApiGatewayToSqsProps);
```

Parámetros

- scopeConstruct
- idstring
- propsApiGatewayToSqsProps

Patrón de construcción

Nombre	Tipo	Descripción
¿ApigatewayProps?	api.RestApiProps	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para API Gateway.
¿Props en cola?	sqs.QueueProps	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para la cola.
DeployDeadLetterQueue?	boolean	Si se va a implementar una cola secundaria para utilizars e como cola de mensajes fallidos. El valor predeterm inado es true.
maxReceiveCount	number	El número de veces que un mensaje se puede quitar sin éxito antes de que se traslade a la cola de mensajes fallidos.
AllowCreateOperation?	boolean	Si desea implementar un método de API Gateway para operaciones de creación en

Nombre	Tipo	Descripción
		la cola (es decir, SQS:SendMessage).
CreateRequestTemplate?	`string`	Anule la plantilla de solicitud de API Gateway predeterminada para el método Create, si `allowCreateOperation` toma el valor `true`.
AllowReadOperation?	`boolean`	Si desea implementar un método de API Gateway para operaciones de lectura en la cola (es decir, SQS:ReceiveMessage).
ReadRequestTemplate?	`string`	Anule la plantilla de solicitud de API Gateway predeterminada para el método Read, si `allowReadOperation` toma el valor `true`.
AllowDeleteOperation?	`boolean`	Si desea implementar un método de API Gateway para operaciones de eliminación en la cola (es decir, SQS:DeleteMessage).
¿DeleteRequestTemplate?	`string`	Anule la plantilla de solicitud de API Gateway predeterminada para el método Delete, si `allowDeleteOperation` toma el valor `true`.

Nombre	Tipo	Descripción
LogGroupProps?	`logs.LogGroupProps`	Servicios opcionales proporcionados por el usuario para anular los accesorios predeterminados para el grupo de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
ApiGateway	`api.RestApi`	Devuelve una instancia de la API REST de API Gateway creada por el patrón.
ApigatewayCloudWatchRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón que permite el registro de acceso desde API Gateway REST API a CloudWatch.
ApigatewayLogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por el patrón al que se envían los registros de acceso de API REST de API de API Gateway de API.
ApigateWayRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón para la API Gateway.
DeadLetterQueue?	`sqs.Queue`	Devuelve una instancia de la cola de letras muertas creada

Nombre	Tipo	Descripción
		por el patrón, si se implementa una.
SQSqueue	`sqs.Queue`	Devuelve una instancia de la cola SQS creada por el patrón.

Ejemplo de uso de las API

Método	Ruta de solicitud	Cuerpo de la solicitud	Acción de cola	Descripción
GET	`/`		`sqs::ReceiveMessage`	Recupera un mensaje de la cola.
POST	`/`	`{ "data": "Hello World!" }`	`sqs::SendMessage`	Entrega un mensaje a la cola.
ELIMINAR	`/message?receiptHandle=[value]`		`sqs::DeleteMessage`	Elimina un mensaje especificado de la cola

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Amazon API Gateway

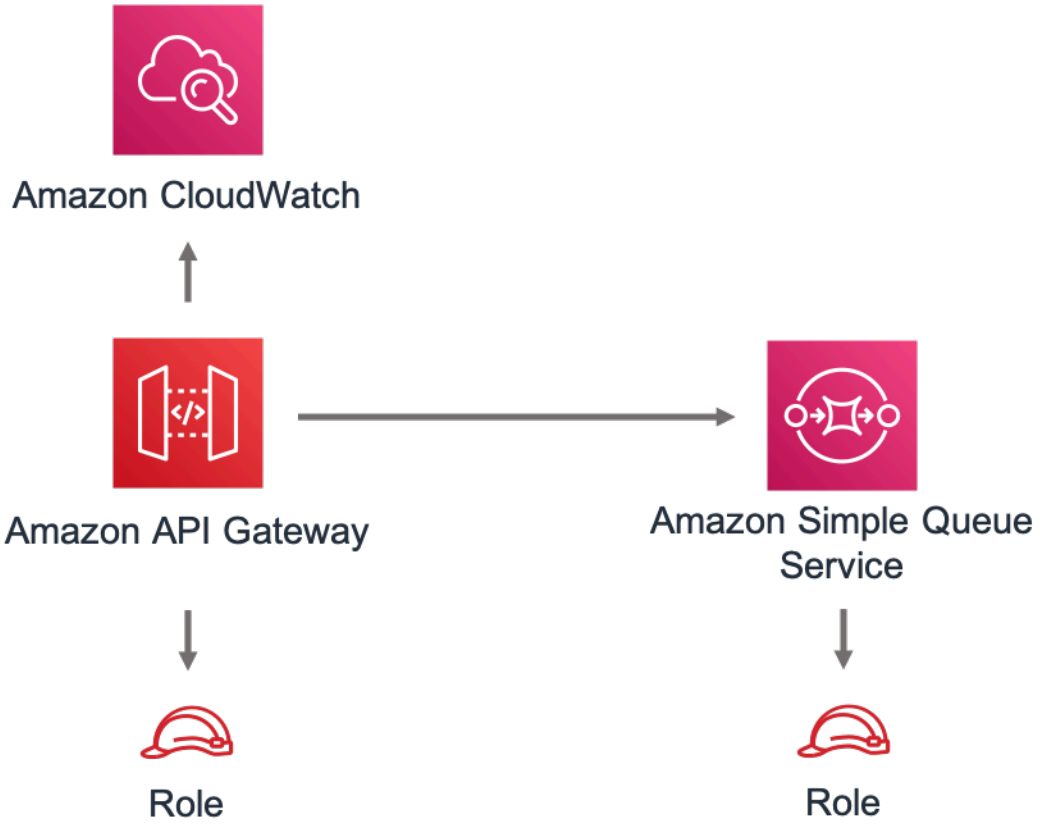
- Implementación de un extremo de API optimizado para bordes
- Habilite el registro de CloudWatch para API Gateway
- Configurar el rol de IAM de acceso mínimo con privilegios para API Gateway

- Establezca el AuthorizationType predeterminado para todos los métodos API en IAM
- Habilite el X-Ray streo

Cola de Amazon SQS

- Implementación de cola de mensajes fallidos de SQS para la cola de SQS de origen
- Habilite el cifrado en el lado de servidor para la cola de SQS de origen con la clave KMS administrada de AWS
- Aplicación del cifrado de los datos en tránsito

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-apigateway-sqs

aws-cloudfront-apigateway

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_cloudfront_apigateway`
TypeScript	`@aws-solutions-constructs/aws-cloudfront-apigateway`
Java	`software.amazon.awsconstructs.services.cloudfrontapigateway`

Overview

Este constructo de soluciones de AWS implementa una distribución de Amazon CloudFront frente a una API REST de Amazon API Gateway.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import * as api from '@aws-cdk/aws-apigateway';
import * as lambda from "@aws-cdk/aws-lambda";
import { CloudFrontToApiGateway } from '@aws-solutions-constructs/aws-cloudfront-
apigateway';
```

```
const lambdaProps: lambda.FunctionProps = {
    code: lambda.Code.fromAsset(`${__dirname}/lambda`),
    runtime: lambda.Runtime.NODEJS_12_X,
    handler: 'index.handler'
};

const lambdafunction = new lambda.Function(this, 'LambdaFunction', lambdaProps);

const apiGatewayProps: api.LambdaRestApiProps = {
        handler: lambdafunction,
        endpointConfiguration: {
            types: [api.EndpointType.REGIONAL]
        },
        defaultMethodOptions: {
            authorizationType: api.AuthorizationType.NONE
        }
};

const apiGateway = new api.LambdaRestApi(this, 'LambdaRestApi', apiGatewayProps);

new CloudFrontToApiGateway(this, 'test-cloudfront-apigateway', {
    existingApiGatewayObj: apiGateway
});
```

Initializer

```
new CloudFrontToApiGateway(scope: Construct, id: string, props:
 CloudFrontToApiGatewayProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`CloudFrontToApiGatewayProps`

Patrón de construcción de construcción

Nombre	Tipo	Descripción
ExistenteApigateWaYobj	`api.RestApi`	La puerta de API Gateway regional que se presentará con CloudFront
CloudFrontDistributionProps?	`cloudfront.DistributionProps`	El usuario ha proporcionado accesorios opcionales para anular los accesorios predeterminados para la distribución de CloudFront.
InsertHTTPSecurityHeaders?	`boolean`	Los apoyos proporcionados por el usuario opcionales para activar/desactivar la inyección automática de cabeceras de seguridad HTTP de las mejores prácticas en todas las respuestas de CloudFront

Propiedades de patrón

Nombre	Tipo	Descripción
ApiGateway	`api.RestApi`	Devuelve una instancia de la API REST de API Gateway creada por el patrón.
CloudFrontLogingBucket?	`s3.Bucket`	Devuelve una instancia del depósito de registro creado por el patrón para la distribución web de CloudFront.

Nombre	Tipo	Descripción
CloudFrontWebDistribution	`cloudfront.CloudFrontWebDistribution`	Devuelve una instancia de la distribución web de CloudFront creada por el patrón.
EdgeLambdaFontionVersion?	`lambda.Version`	Devuelve una instancia de la versión de la función de borde de Lambda creada por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

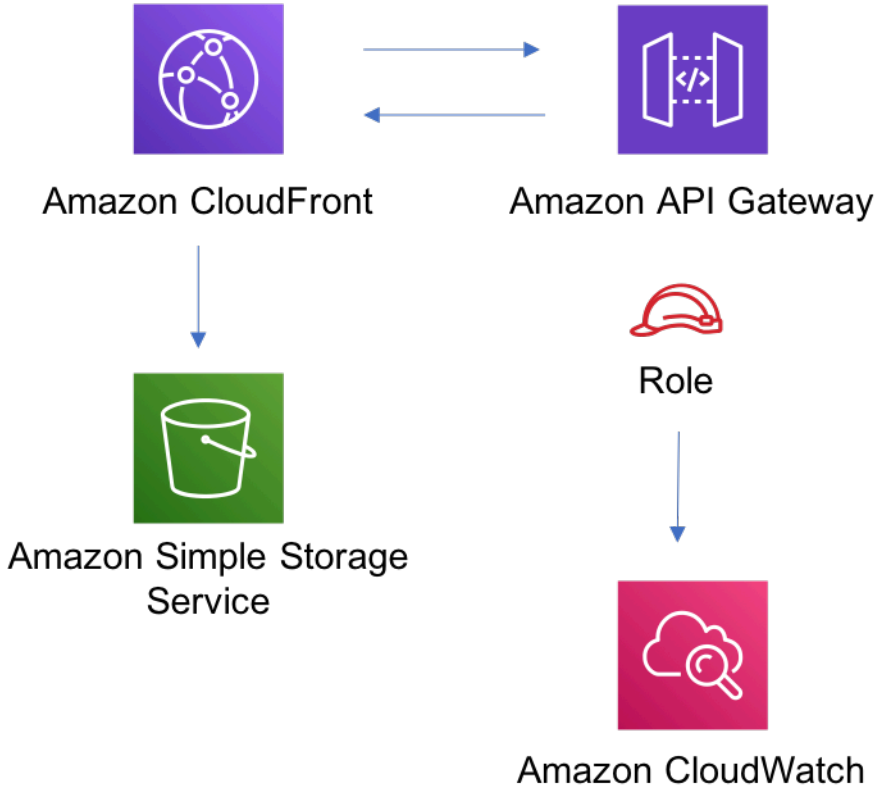
Amazon CloudFront

- Configurar el registro de acceso para CloudFront WebDistribution
- Habilitar la inyección automática de encabezados de seguridad HTTP de mejores prácticas en todas las respuestas de CloudFront WebDistribution

Amazon API Gateway

- El objeto API Gateway proporcionado por el usuario se utiliza tal cual
- Habilitar el X-Ray streo

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-cloudfront-apigateway

aws-cloudfront-apigateway-lambda

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersionado semánticoEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_cloudfront_apigateway_lambda`
Mecanografía	`@aws-solutions-constructs/aws-cloudfront-apigateway-lambda`
Java	`software.amazon.awsconstructs.services.cloudfrontapigatewaylambda`

Overview

Este Construct de soluciones de AWS implementa una distribución de Amazon CloudFront frente a una API REST respaldada por Lambda de Amazon API Gateway.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { CloudFrontToApiGatewayToLambda } from '@aws-solutions-constructs/aws-
cloudfront-apigateway-lambda';

new CloudFrontToApiGatewayToLambda(this, 'test-cloudfront-apigateway-lambda', {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    }
});
```

Initializer

```
new CloudFrontToApiGatewayToLambda(scope: Construct, id: string, props:
 CloudFrontToApiGatewayToLambdaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`CloudFrontToApiGatewayToLambdaProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si se`existingLambdaObj` se encuentra.
¿ApigatewayProps?	`api.LambdaRestApiProps`	Los apoyos proporcionados por el usuario opcionales para anular los apoyos predeterminados para API Gateway

Nombre	Tipo	Descripción
CloudFrontDistributionProps?	`cloudfront.DistributionProps`	El usuario ha proporcionado accesorios opcionales para anular los accesorios predeterminados para la distribución de CloudFront.
InsertHTTPSecurityHeaders?	`boolean`	Los apoyos proporcionados por el usuario opcionales para activar/desactivar la inyección automática de encabezados de seguridad HTTP de las mejores prácticas en todas las respuestas de CloudFront
LogGroupProps?	`logs.LogGroupProps`	Los apoyos opcionales proporcionados por el usuario para anular los apoyos predeterminados para el grupo de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
ApiGateway	`api.RestApi`	Devuelve una instancia de API Gateway creada por el patrón.
ApigatewayCloudWatchRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón que permite el registro de acceso desde API Gateway REST API a CloudWatch.

Nombre	Tipo	Descripción
ApigatewayLogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por el patrón al que se envían los registros de acceso de API REST de API de API Gateway de API.
CloudFrontLogingBucket?	`s3.Bucket`	Devuelve una instancia del depósito de registro creado por el patrón para la distribución web de CloudFront.
CloudFrontWebDistribution	`cloudfront.CloudFrontWebDistribution`	Devuelve una instancia de la distribución web de CloudFront creada por el patrón.
EdgelambdaFontionVersion?	`lambda.Version`	Devuelve una instancia de la versión de la función de borde de Lambda creada por el patrón.
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Amazon CloudFront

- Configurar el registro de acceso para CloudFront WebDistribution
- Habilitar la inyección automática de encabezados de seguridad HTTP de mejores prácticas en todas las respuestas de CloudFront WebDistribution

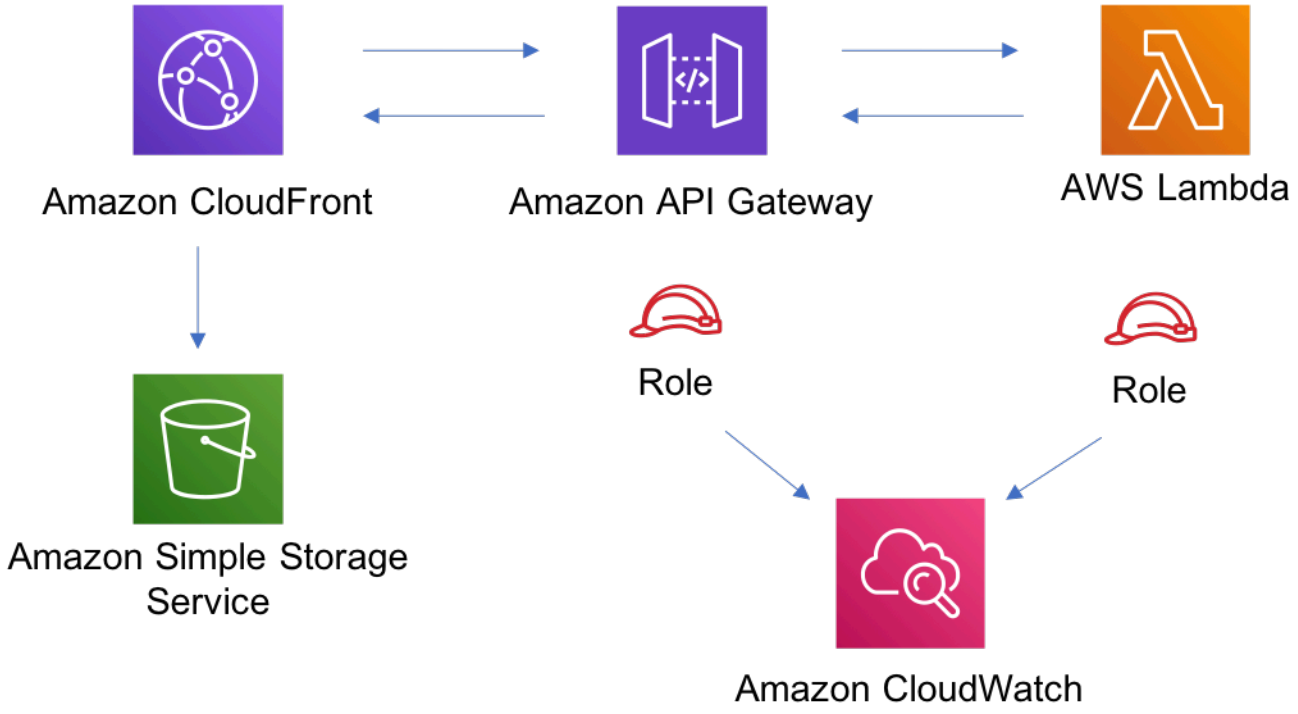
Amazon API Gateway

- Implementar un extremo de API regional
- Habilitar el registro de CloudWatch para API Gateway
- Configurar el rol de IAM de acceso mínimo con privilegios para API Gateway
- Establezca el AuthorizationType predeterminado para todos los métodos API en IAM
- Habilitar el X-Ray streo

Función de AWS Lambda

- Configuración de la función de IAM de acceso limitado con privilegios para la función Lambda
- Habilitar conexiones de reutilización con la función Keep-Alive para NodeJS Lambda
- Habilitar el X-Ray streo
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-cloudfront-apigateway-lambda

aws-cloudfront-mediastore

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersionado semánticoModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_cloudfront_mediastore`
Mecanografiado	`@aws-solutions-constructs/aws-cloudfront-mediastore`
Java	`software.amazon.awsconstructs.services.cloudfrontmediastore`

Overview

Este constructo de soluciones de AWS implementa una distribución de Amazon CloudFront conectada a un contenedor AWS Elemental MediaStore.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { CloudFrontToMediaStore } from '@aws-solutions-constructs/aws-cloudfront-
mediastore';

new CloudFrontToMediaStore(this, 'test-cloudfront-mediastore-default', {});
```

Initializer

```
new CloudFrontToMediaStore(scope: Construct, id: string, props:
 CloudFrontToMediaStoreProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`CloudFrontToMediaStoreProps`

Patrón de construcción

Nombre	Tipo	Descripción
ExistenteMediaStoreContainerObj?	`mediastore.CfnContainer`	Contenedor MediaStore opcional proporcionado por el usuario para anular el contenedor MediaStore predeterminado.
MediaStoreContainerProps?	`mediastore.CfnContainerProps`	Props opcionales proporcionados por el usuario para anular los apoyos predeterminados para el Contenedor MediaStore.

Nombre	Tipo	Descripción
CloudFrontDistributionProps?	cloudfront.DistributionProps | any	Props opcionales proporcionados por el usuario para anular los apoyos predeterminados para la distribución de CloudFront.
InsertHTTPSecurityHeaders?	boolean	Props opcionales proporcionados por el usuario para activar o desactivar la inyección automática de encabezados de seguridad HTTP de mejores prácticas en todas las respuestas de CloudFront.

Propiedades de patrón

Nombre	Tipo	Descripción
CloudFrontWebDistribution	cloudfront.CloudFrontWebDistribution	Devuelve una instancia de la distribución web de CloudFront creada por el patrón.
MediaStoreContainer	mediastore.CfnContainer	Devuelve una instancia del contenedor MediaStore creado por el patrón.
CloudFrontLogingBucket	s3.Bucket	Devuelve una instancia del depósito de registro creado por el patrón para la distribución web de CloudFront.
CloudFrontOriginRequestPolicy	cloudfront.OriginRequestPolicy	Devuelve una instancia de la directiva de solicitud de origen de CloudFront creada por el

Nombre	Tipo	Descripción
		patrón para la distribución web de CloudFront.
Propiedad CloudFrontOriginAccessIdentiAccessIdentity	`cloudfront.OriginAccessIdentity`	Devuelve una instancia de la identidad de acceso de origen de CloudFront creada por el patrón para la distribución web de CloudFront.
EdgeLambdaFontionVersion	`lambda.Version`	Devuelve una instancia de la versión de la función de borde de Lambda creada por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Amazon CloudFront

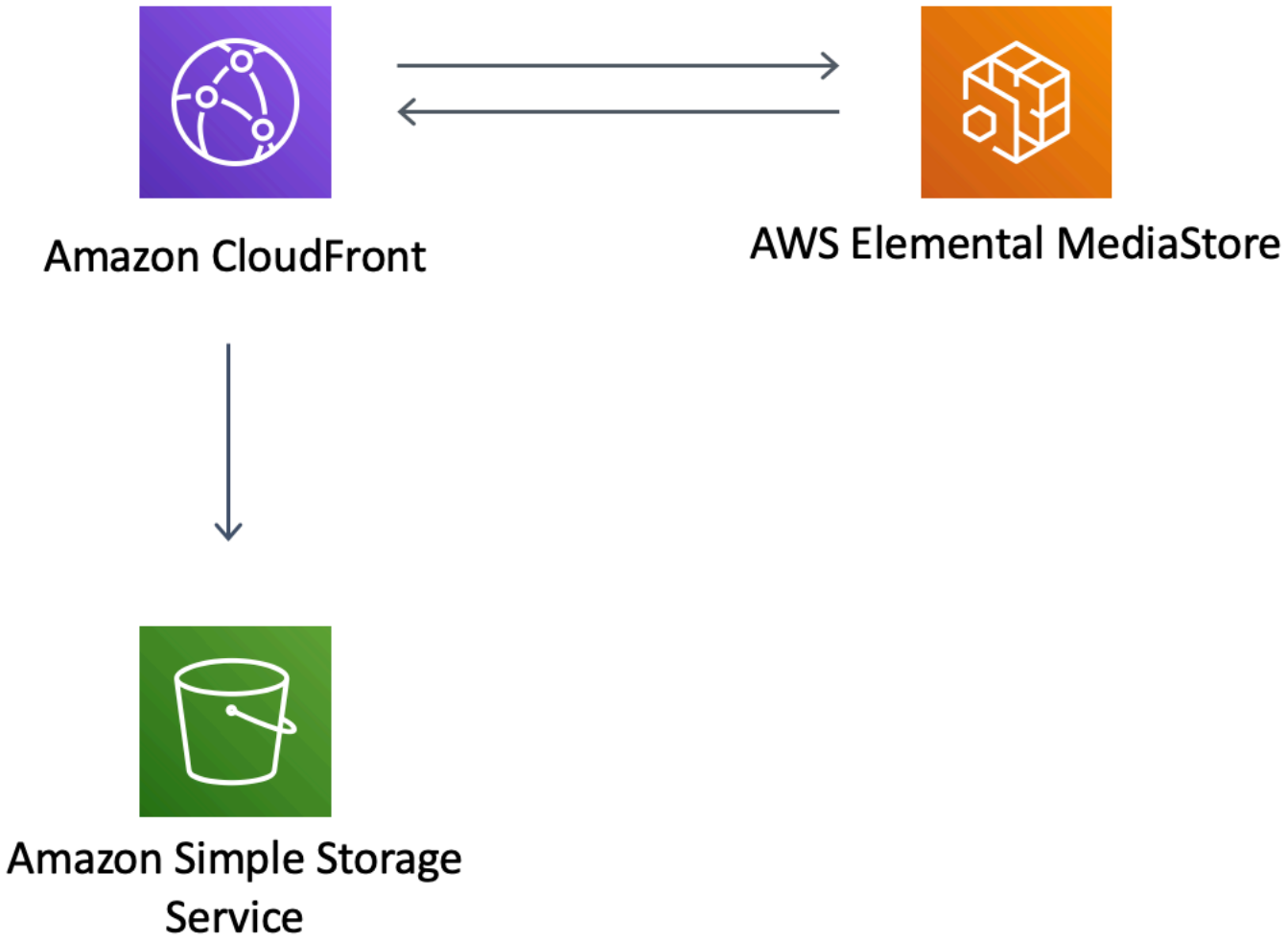
- Configuración del registro de acceso para la distribución web de CloudFront
- Habilitar la política de solicitud de origen de CloudFront para el contenedor AWS Elemental MediaStore
- Establezca`User-Agent`Propiedad personalizada con identidad de acceso de origen de CloudFront
- Habilitar la inyección automática de encabezados de seguridad HTTP de mejores prácticas en todas las respuestas de la distribución web de CloudFront

AWS Elemental MediaStore

- Establecer la directiva de eliminación para conservar el recurso
- Establecer el nombre del contenedor con el nombre de la pila CloudFormation
- Configure el predeterminadoPropiedad compartida de recursos entre orígenes (CORS)
- Configure el predeterminadoPolítica de ciclo de vida de los objetos

- Configure el predeterminadoPolítica de contenedorpara permitir sólo`aws:UserAgent`identidad de acceso de origen de CloudFront
- Configure el predeterminadoPolítica de métricas
- Habilitar el registro de acceso

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-cloudfront-mediastore

aws-cloudfront-s3

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_cloudfront_s3`
Mecanografía	`@aws-solutions-constructs/aws-cloudfront-s3`
Java	`software.amazon.awsconstructs.services.cloudfronts3`

Overview

Este constructo de soluciones de AWS implementa una distribución de Amazon CloudFront frente a un bucket de Amazon S3.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { CloudFrontToS3 } from '@aws-solutions-constructs/aws-cloudfront-s3';

new CloudFrontToS3(this, 'test-cloudfront-s3', {});
```

Initializer

```
new CloudFrontToS3(scope: Construct, id: string, props: CloudFrontToS3Props);
```

Parámetros

- scope`Construct`
- id`string`
- props`CloudFrontToS3Props`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteBuckeTobj?	`s3.Bucket`	Instancia existente del objeto S3 Bucket. Si se proporcio na esto, entonces también proporcionabucketProps es un error.
¿BucketProps?	`s3.BucketProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas del depósito. Se pasa por alto si unexistingBucketObj se encuentra.
CloudFrontDistributionProps?	`cloudfront.DistributionProps`	El usuario ha proporcio nado accesorios opcionale s para anular los accesorio s predeterminados para la distribución de CloudFront.

Nombre	Tipo	Descripción
InsertHTTPSecurityHeaders?	`boolean`	Los apoyos proporcionados por el usuario opcionales para activar o desactivar la inyección automática de encabezados de seguridad HTTP de las mejores prácticas en todas las respuestas de CloudFront

Propiedades de patrón

Nombre	Tipo	Descripción
CloudFrontWebDistribution	`cloudfront.CloudFrontWebDistribution`	Devuelve una instancia de la distribución web de CloudFront creada por el patrón.
S3 Bucket?	`s3.Bucket`	Devuelve una instancia del bucket S3 creado por el patrón.
¿S3LoggingBucket?	`s3.Bucket`	Devuelve una instancia del depósito de registro creado por el patrón para el bucket S3.
EdgeLambdaFontionVersion?	`lambda.Version`	Devuelve una instancia de la versión de la función de borde de Lambda creada por el patrón.
CloudFrontLogingBucket?	`s3.Bucket`	Devuelve una instancia del depósito de registro creado

Nombre	Tipo	Descripción
		por el patrón para la distribución web de CloudFront.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

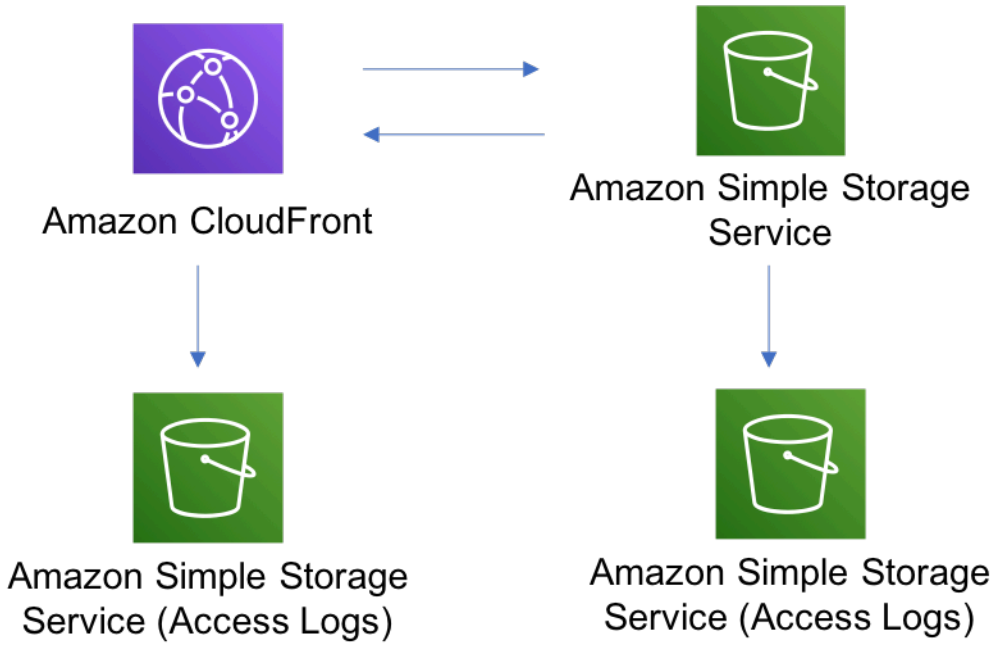
Amazon CloudFront

- Configurar el registro de acceso para CloudFront WebDistribution
- Habilitar la inyección automática de encabezados de seguridad HTTP de mejores prácticas en todas las respuestas de CloudFront WebDistribution

Bucket de Amazon S3

- Configurar el registro de acceso para S3 Bucket
- Habilitar el cifrado del lado del servidor para S3 Bucket utilizando la clave KMS administrada por AWS
- Activar el control de versiones para S3 Bucket
- No permitir el acceso público para S3 Bucket
- Conservar el depósito de S3 al eliminar la pila de CloudFormation
- Aplicación del cifrado de los datos en tránsito
- Aplica una regla de ciclo de vida para mover versiones de objetos no actuales al almacenamiento de Glacier después de 90 días

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-cloudfront-s3

aws-cognito-apigateway-lambda

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_cognito_apigateway_lambda`
Mecanografía	`@aws-solutions-constructs/aws-cognito-apigateway-lambda`
Java	`software.amazon.awsconstructs.services.cognitoapigatewaylambda`

Overview

Este Construct de soluciones de AWS implementa Amazon Cognito protegiendo una API REST respaldada por Lambda de Amazon API Gateway.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { CognitoToApiGatewayToLambda } from '@aws-solutions-constructs/aws-cognito-apigateway-lambda';

new CognitoToApiGatewayToLambda(this, 'test-cognito-apigateway-lambda', {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    }
});
```

Si está definiendo recursos y métodos en su API (por ejemplo,`proxy = false`), debe llamar a la`addAuthorizers()`después de que la API esté completamente definida. Esto garantiza que todos los métodos de su API estén protegidos.

A continuación se muestra un ejemplo en TypeScript:

```
import { CognitoToApiGatewayToLambda } from '@aws-solutions-constructs/aws-cognito-
apigateway-lambda';

const construct = new CognitoToApiGatewayToLambda(this, 'test-cognito-apigateway-
lambda', {
    lambdaFunctionProps: {
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        runtime: lambda.Runtime.NODEJS_12_X,
        handler: 'index.handler'
    },
    apiGatewayProps: {
      proxy: false
    }
});

const resource = construct.apiGateway.root.addResource('foobar');
resource.addMethod('POST');

// Mandatory to call this method to Apply the Cognito Authorizers on all API methods
construct.addAuthorizers();
```

Initializer

```
new CognitoToApiGatewayToLambda(scope: Construct, id: string, props:
 CognitoToApiGatewayToLambdaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`CognitoToApiGatewayToLambdaProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como `lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si `unexistingLambdaObj` se encuentra.
¿ApigateWayProps?	`api.LambdaRestApiProps`	Los apoyos proporcionados por el usuario opcionales para anular los apoyos predeterminados para API Gateway
¿CognitouserPoolProps?	`cognito.UserPoolProps`	Los accesorios proporcionados por el usuario opcionales para anular los accesorios predeterminados para el grupo de usuarios de Cognito
¿CognitouserPoolClientProps?	`cognito.UserPoolClientProps`	Los accesorios proporcionados por el usuario opcionales para anular los accesorios predeterminados para Cognito User Pool Client
LogGroupProps?	`logs.LogGroupProps`	Los apoyos opcionales proporcionados por el usuario

Nombre	Tipo	Descripción
		para anular los apoyos predeterminados para el grupo de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
ApiGateway	`api.RestApi`	Devuelve una instancia de la API REST de API Gateway creada por el patrón.
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.
userPool	`cognito.UserPool`	Devuelve una instancia del grupo de usuarios de Cognito creado por el patrón.
UserPoolClient	`cognito.UserPoolClient`	Devuelve una instancia del cliente de grupo de usuarios de Cognito creado por el patrón.
ApigatewayCloudWatchRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón que permite el registro de acceso desde API Gateway REST API a CloudWatch.
ApigatewayLogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por el patrón al que se envían los

Nombre	Tipo	Descripción
		registros de acceso de API REST de API de API Gateway de API.
ApiGateWayAuthorizer	`api.CfnAuthorizer`	Devuelve una instancia del autorizador de API Gateway creado por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Amazon Cognito

- Configuración de directiva de contraseñas para grupos de usuarios de
- Aplicar el modo de seguridad avanzada para grupos de usuarios

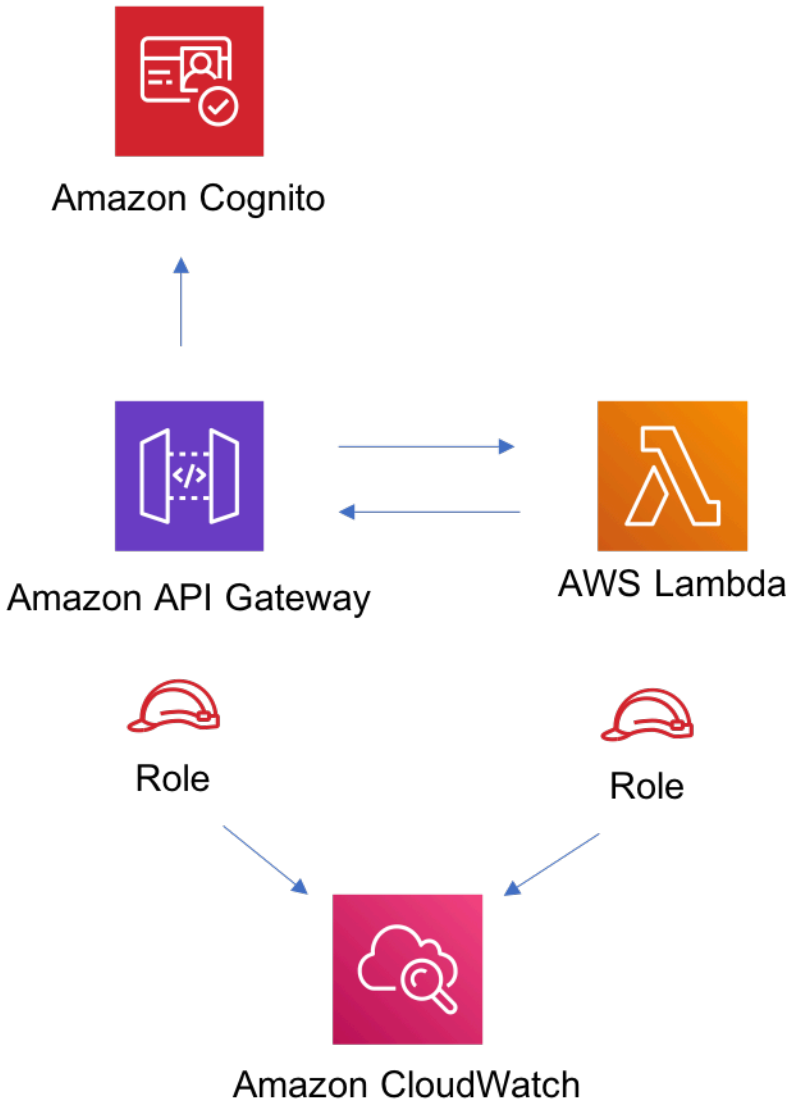
Amazon API Gateway

- Implementación de un extremo de API optimizado para bordes
- Habilitar el registro de CloudWatch para API Gateway
- Configurar el rol de IAM de acceso mínimo con privilegios para API Gateway
- Establezca el AuthorizationType predeterminado para todos los métodos API en IAM
- Habilitar el X-Ray streo

Función de AWS Lambda

- Configuración de la función de IAM de acceso limitado a privilegios para la función Lambda
- Habilitar conexiones de reutilización con la función Keep-Alive para NodeJS Lambda
- Habilitar el X-Ray streo
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-cognito-apigateway-lambda

aws-dynamodb-stream-lambda

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersionado semánticoModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_dynamodb_stream_lambda`
Mecanografía	`@aws-solutions-constructs/aws-dynamodb-stream-lambda`
Java	`software.amazon.awsconstructs.services.dynamodbstreamlambda`

Overview

Este constructo de soluciones de AWS implementa un patrón de tabla de Amazon DynamoDB con flujo para invocar la función AWS Lambda con los permisos menos privilegiados.

Aquí hay una definición mínima de patrón implementable:

```
import { DynamoDBStreamToLambdaProps, DynamoDBStreamToLambda} from '@aws-solutions-constructs/aws-dynamodb-stream-lambda';

new DynamoDBStreamToLambda(this, 'test-dynamodb-stream-lambda', {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
```

```
    },
});
```

Initializer

```
new DynamoDBStreamToLambda(scope: Construct, id: string, props:
 DynamoDBStreamToLambdaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`DynamoDBStreamToLambdaProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si un`existingLambdaObj` se encuentra a su disposición.
¿DynamotableProps?	`dynamodb.TableProps`	Los accesorios proporcionados por el usuario opcionale

Nombre	Tipo	Descripción
		s para anular los accesorios predeterminados para la tabla de DynamoDB
ExistingTableObj?	`dynamodb.Table`	Instancia existente del objeto de tabla DynamoDB, proporcionando tanto esto como`dynamoTableProps` provocará un error.
DynamoEventSourceProps?	`aws-lambda-event-sources.DynamoEventSourceProps`	Los apoyos proporcionados por el usuario opcionales para anular los apoyos predeterminados para el origen de eventos de DynamoDB

Propiedades de patrón

Nombre	Tipo	Descripción
DynamoTable	`dynamodb.Table`	Devuelve una instancia de la tabla DynamoDB creada por el patrón.
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.

Lambda function

Este patrón requiere una función Lambda que pueda publicar datos en el servicio Elasticsearch desde la secuencia de DynamoDB. Se proporciona una función de muestraAquí.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

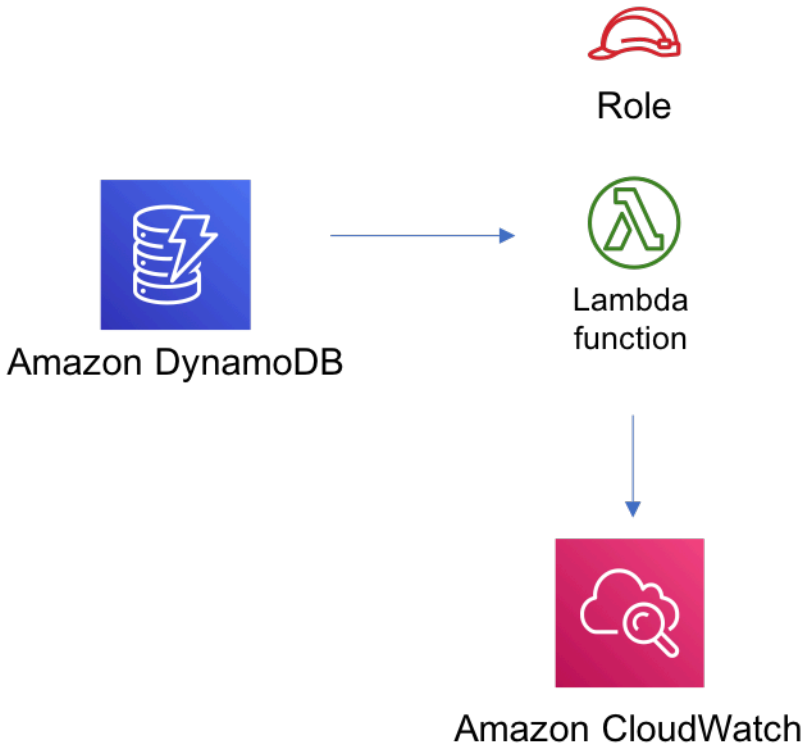
Tabla Amazon DynamoDB

- Establecer el modo de facturación para la tabla de DynamoDB en Bajo demanda (pago por solicitud)
- Habilitar el cifrado del lado del servidor para DynamoDB Table mediante la clave KMS administrada por AWS
- Crea una clave de partición llamada 'id' para la tabla de DynamoDB
- Conservar la tabla al eliminar la pila CloudFormation
- Permita las copias de seguridad continuas y la recuperación a un momento dado

Función de AWS Lambda

- Configuración de la función IAM de acceso a privilegios limitados para la función Lambda
- Habilitar conexiones de reutilización con la función Keep-Alive para NodeJS Lambda
- Habilitar el X-Ray streo
- Habilitar funciones de control de errores: habilitar bisect en función Error; establecer la antigüedad máxima de registro predeterminada (24 horas); establecer intentos máximos de reintento (500) por defecto; e implementar la cola de letras muertas SQS como destino en caso de fallo
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-dynamodb-stream-lambda

aws-dynamodb-stream-lambda-elasticsearch-kibana

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersioning semánticoEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_dynamodb_stream_lambda_elasticsearch_kibana`
Mecanografiado	`@aws-solutions-constructs/aws-dynamodb-stream-lambda-elasticsearch-kibana`
Java	`software.amazon.awsconstructs.services.dynamodbstreamlambdaelasticsearchkibana`

Overview

Este Construct de soluciones de AWS implementa una tabla de Amazon DynamoDB con flujo, una función de AWS Lambda y un Amazon Elasticsearch Service con los permisos menos privilegiados.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { DynamoDBStreamToLambdaToElasticSearchAndKibana,
 DynamoDBStreamToLambdaToElasticSearchAndKibanaProps } from '@aws-solutions-constructs/
aws-dynamodb-stream-lambda-elasticsearch-kibana';
import { Aws } from "@aws-cdk/core";

const props: DynamoDBStreamToLambdaToElasticSearchAndKibanaProps = {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    },
    domainName: 'test-domain',
    // TODO: Ensure the Cognito domain name is globally unique
    cognitoDomainName: 'globallyuniquedomain' + Aws.ACCOUNT_ID;
};
```

```
new DynamoDBStreamToLambdaToElasticSearchAndKibana(this, 'test-dynamodb-stream-lambda-elasticsearch-kibana', props);
```

Initializer

```
new DynamoDBStreamToLambdaToElasticSearchAndKibana(scope: Construct, id: string, props: DynamoDBStreamToLambdaToElasticSearchAndKibanaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`DynamoDBStreamToLambdaToElasticSearchAndKibanaProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si `unexistingLambdaObj` se encuentra.
¿DynamotableProps?	`dynamodb.TableProps`	Los accesorios proporcionados por el usuario opcionale

Nombre	Tipo	Descripción
		s para anular los accesorios predeterminados para la tabla de DynamoDB
ExistingTableObj?	`dynamodb.Table`	Instancia existente del objeto de tabla DynamoDB, proporcionando tanto esto como`dynamoTableProps` provocará un error.
DynamoEventSourceProps?	`aws-lambda-event-sources.DynamoEventSourceProps`	Los apoyos proporcionados por el usuario opcionales para anular los apoyos predeterminados para el origen de eventos de DynamoDB
¿EsDomainProps?	`elasticsearch.CfnDomainProps`	Los accesorios proporcionados por el usuario opcionales para anular los accesorios predeterminados para Amazon Elasticsearch Service
domainName	`string`	Nombre de dominio para Cognito y Amazon Elasticsearch Service
CreateCloudWatchalarms	`boolean`	Si desea crear alarmas recomendadas de CloudWatch.

Propiedades de patrón

Nombre	Tipo	Descripción
¿Cloudwatchalarms?	`cloudwatch.Alarm[]`	Devuelve una lista de una o más alarmas de CloudWatch creadas por el patrón.
DynamoTable	`dynamodb.Table`	Devuelve una instancia de la tabla DynamoDB creada por el patrón.
ElasticSearchDomain	`elasticsearch.CfnDomain`	Devuelve una instancia del dominio Elasticsearch creado por el patrón.
IdentityPool	`cognito.CfnIdentityPool`	Devuelve una instancia del grupo de identidades de Cognito creado por el patrón.
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.
userPool	`cognito.UserPool`	Devuelve una instancia del grupo de usuarios de Cognito creado por el patrón.
UserPoolClient	`cognito.UserPoolClient`	Devuelve una instancia del cliente de grupo de usuarios de Cognito creado por el patrón.

Lambda function

Este patrón requiere una función Lambda que pueda publicar datos en el servicio Elasticsearch desde la secuencia de DynamoDB. Se proporciona una función de muestraAquí.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Tabla Amazon DynamoDB

- Establezca el modo de facturación para la tabla de DynamoDB en Bajo demanda (pago por solicitud)
- Habilitar el cifrado del lado del servidor para DynamoDB Table mediante la clave KMS administrada por AWS
- Crea una clave de partición llamada 'id' para la tabla de DynamoDB
- Conservar la tabla al eliminar la pila CloudFormation
- Habilite las copias de seguridad continuas y la recuperación a un momento dado

Función de AWS Lambda

- Configuración de la función IAM de acceso a privilegios limitados para la función Lambda
- Habilitar conexiones de reutilización con la función Keep-Alive para NodeJS Lambda
- Habilitar el X-Ray streo
- Habilitar funciones de control de errores: habilitar bisect en función Error; establecer la antigüedad máxima de registro predeterminada (24 horas); establecer intentos máximos de reintento (500) por defecto; e implementar la cola de letras muertas SQS como destino en caso de fallo
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Amazon Cognito

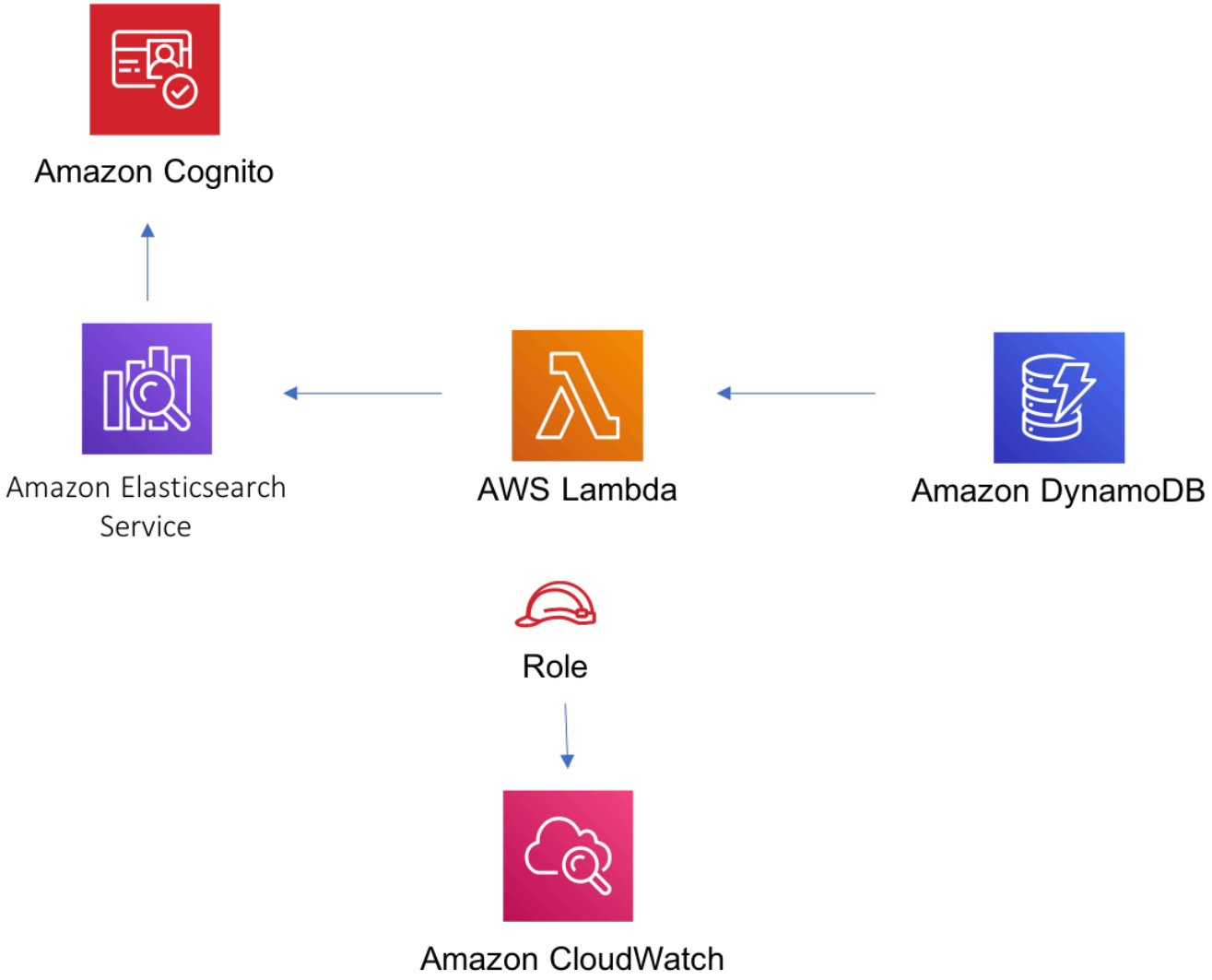
- Configuración de la directiva de contraseñas para grupos de usuarios
- Aplicar el modo de seguridad avanzada para grupos de usuarios

Amazon Elasticsearch Service

- Implementar las mejores prácticas de las alarmas de CloudWatch para el dominio de Elasticsearch
- Proteja el acceso al panel de control de Kibana Cognito grupos de usuarios de
- Habilitar el cifrado del lado del servidor para Elasticsearch Domain mediante la clave KMS administrada por AWS

- Habilitar el cifrado de nodo a nodo para el dominio Elasticsearch
- Configuración del clúster para el dominio Amazon ES

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	[@aws -soluciones-constructs/aws-dynamodb-stream-lambda-elasticsearch-kibana]

aws-events-rule-kinesisfiremanse-s3

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasmodelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_events_rule_kinesisfirehose_s3`
Mecanografía	`@aws-solutions-constructs/aws-events-rule-kinesisfirehose-s3`
Java	`software.amazon.awsconstructs.services.eventsrulekinesisfirehoses3`

Overview

Este constructo de soluciones de AWS implementa una regla de Amazon CloudWatch Events para enviar datos a una secuencia de entrega de Amazon Kinesis Data Firehose conectada a un bucket de Amazon S3.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import * as cdk from '@aws-cdk/core';
import { EventsRuleToKinesisFirehoseToS3, EventsRuleToKinesisFirehoseToS3Props } from
 '@aws-solutions-constructs/aws-events-rule-kinesisfirehose-s3';
```

```
const eventsRuleToKinesisFirehoseToS3Props: EventsRuleToKinesisFirehoseToS3Props = {
    eventRuleProps: {
    schedule: events.Schedule.rate(cdk.Duration.minutes(5))
    }
};

new EventsRuleToKinesisFirehoseToS3(this, 'test-events-rule-firehose-s3',
 eventsRuleToKinesisFirehoseToS3Props);
```

Initializer

```
new EventsRuleToKinesisFirehoseToS3(scope: Construct, id: string, props:
 EventsRuleToKinesisFirehoseToS3Props);
```

Parámetros

- scope`Construct`
- id`string`
- props`EventsRuleToKinesisFirehoseToS3Props`

Patrón de construcción

Nombre	Tipo	Descripción
EventRuleProps	`events.RuleProps`	Propiedades proporcionadas por el usuario para anular las propiedades predeterminadas de la regla CloudWatch Events.
¿KinesisFireHoseProps?	`aws-kinesisfirehose.CfnDeliveryStreamProps`	El usuario ha proporcionado accesorios opcionales para anular los accesorios predeterminados para Kinesis Firehose Delivery Stream.

Nombre	Tipo	Descripción
¿ExistenteBuckeTobj?	`s3.IBucket`	Instancia existente del objeto S3 Bucket. Si se proporcio na esto, entonces también proporcionabucketProps es un error.
¿BucketProps?	`s3.BucketProps`	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para el bucket S3.
LogGroupProps?	`logs.LogGroupProps`	Props opcionales proporcio nados por el usuario para anular los accesorios predeterminados para el grupo de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
Regla de eventos	`events.Rule`	Devuelve una instancia de la regla Eventos creada por el patrón.
KinesisFirehose	`kinesisfirehose.CfnDeliveryStream`	Devuelve una instancia de la secuencia de entrega de Kinesis Firehose creada por el patrón.
Bucket S3	`s3.Bucket`	Devuelve una instancia del bucket S3 creado por el patrón.

Nombre	Tipo	Descripción
¿S3LoggingBucket?	`s3.Bucket`	Devuelve una instancia del depósito de registro creado por el patrón para el bucket S3.
¿EventsRole?	`iam.Role`	Devuelve una instancia del rol creado por el componente fijo para la regla CloudWatch Events.
KinesisFireHoserole	`iam.Role`	Devuelve una instancia del rol de IAM creado por el patrón para la secuencia de entrega de Kinesis Firehose.
KinesisFireHoselogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por el patrón al que se envían los registros de acceso de Kinesis Firehose.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Reglas de Amazon CloudWatch Events

- Configure el rol de IAM de acceso mínimo con privilegios para que la regla de eventos se publique en Kinesis Firehose Delivery Stream.

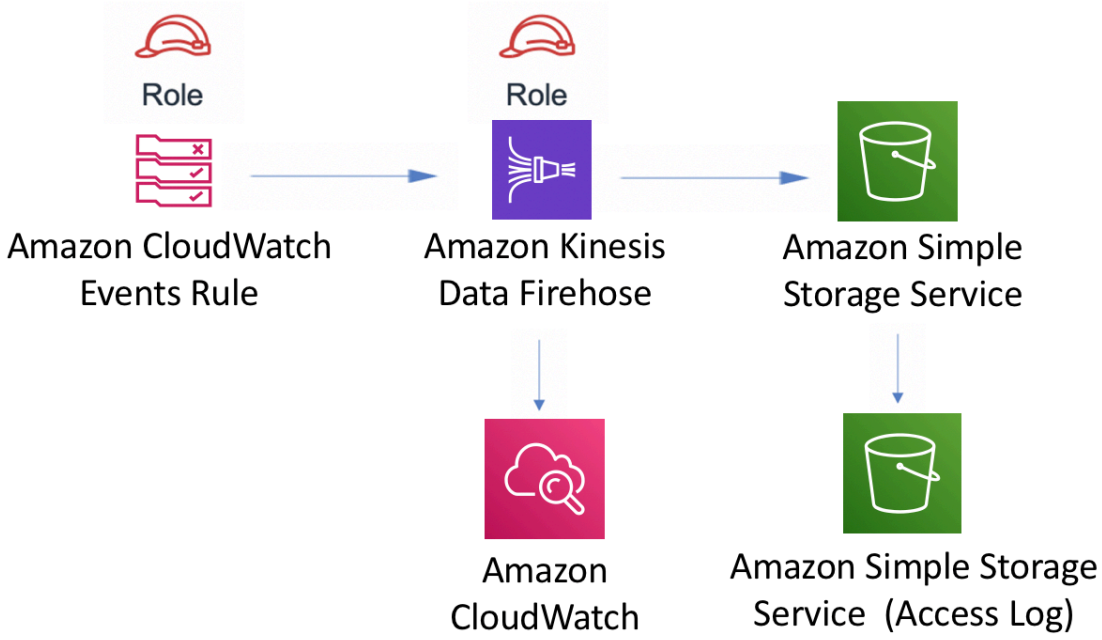
Amazon Kinesis Firehose

- Habilite el registro de CloudWatch para Kinesis Firehose.
- Configurar el rol de IAM de acceso mínimo para Amazon Kinesis Firehose.

Bucket de Amazon S3

- Configurar el registro de acceso para el depósito.
- Habilite el cifrado del lado del servidor para el depósito mediante la clave KMS administrada por AWS.
- Activar el control de versiones para el bucket.
- No permita el acceso público para el cubo.
- Mantenga el depósito al eliminar la pila CloudFormation.
- Aplica una regla de ciclo de vida para mover versiones de objetos no actuales al almacenamiento de Glacier después de 90 días.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-events-rule-kinesisfirehose-s3

aws-events-rule-kinesisstreams

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasmodelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_events_rule_kinesisstream`
Mecanografía	`@aws-solutions-constructs/aws-events-rule-kinesisstreams`
Java	`software.amazon.awsconstructs.services.eventsrulekinesisstream`

Overview

Este constructo de soluciones de AWS implementa una regla de Amazon CloudWatch Events para enviar datos a un flujo de datos de Amazon Kinesis.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import * as cdk from '@aws-cdk/core';
import {EventsRuleToKinesisStreams, EventsRuleToKinesisStreamsProps} from "@aws-
solutions-constructs/aws-events-rule-kinesisstreams";

const props: EventsRuleToKinesisStreamsProps = {
    eventRuleProps: {
      schedule: events.Schedule.rate(Duration.minutes(5)),
    }
};

new EventsRuleToKinesisStreams(this, 'test-events-rule-kinesis-stream', props);
```

Initializer

```
new EventsRuleToKinesisStreams(scope: Construct, id: string, props:
 EventsRuleToKinesisStreamsProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`EventsRuleToKinesisStreamsProps`

Patrón de construcción

Nombre	Tipo	Descripción
EventRuleProps	`events.RuleProps`	Propiedades proporcionadas por el usuario para anular las propiedades predeterm inadas de la regla CloudWatch Events.

Nombre	Tipo	Descripción
¿ExistenteStreamObj?	`kinesis.Stream`	Instancia existente de Kinesis Stream, proporcionando tanto esto como`kinesisStreamProps` Si se produce un error.
¿KinesisStreamProps?	`kinesis.StreamProps`	Props opcionales proporcionados por el usuario para anular los apoyos predeterminados para la transmisión de Kinesis.
CreateCloudWatchalarms	`boolean`	Si desea crear alarmas recomendadas de CloudWatch.

Patrón Propiedades

Nombre	Tipo	Descripción
Regla de eventos	`events.Rule`	Devuelve una instancia de la regla Eventos creada por el patrón.
Flujos de Kinesi	`kinesis.Stream`	Devuelve una instancia de la secuencia Kinesis creada por el patrón.
EventsRole?	`iam.Role`	Devuelve una instancia del rol creado por el componente fijo para la regla CloudWatch Events.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Amazon CloudWatch Events

- Configure el rol de IAM de acceso mínimo con privilegios para que la regla de eventos se publique en el flujo de datos de Kinesis.

Amazon Kinesis Stream

- Habilite el cifrado del lado del servidor para Kinesis Data Stream mediante AWS Managed KMS Key.

Architecture

GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-events-rule-kinesisstreams

aws-events-regla-lambda

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersionado semánticoEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_events_rule_lambda`
Mecanografía	`@aws-solutions-constructs/aws-events-rule-lambda`
Java	`software.amazon.awsconstructs.services.eventsrulelambda`

Overview

Este construcción de soluciones de AWS implementa una regla de eventos de AWS y una función de AWS Lambda.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
const { EventsRuleToLambdaProps, EventsRuleToLambda } from '@aws-solutions-constructs/
aws-events-rule-lambda';

const props: EventsRuleToLambdaProps = {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    },
    eventRuleProps: {
      schedule: events.Schedule.rate(Duration.minutes(5))
    }
};

new EventsRuleToLambda(this, 'test-events-rule-lambda', props);
```

Initializer

```
new EventsRuleToLambda(scope: Construct, id: string, props: EventsRuleToLambdaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`EventsRuleToLambdaProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function,

Nombre	Tipo	Descripción
		proporcionando tanto esto como `lambdaFunctionProps` provocará un error.
LambdaFontionProps	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si `unexistingLambdaObj` se encuentra.
EventRuleProps	`events.RuleProps`	El usuario proporcionó EventRuleProps para anular los valores predeterminados

Propiedades de patrón

Nombre	Tipo	Descripción
Regla de eventos	`events.Rule`	Devuelve una instancia de la regla Eventos creada por el patrón.
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

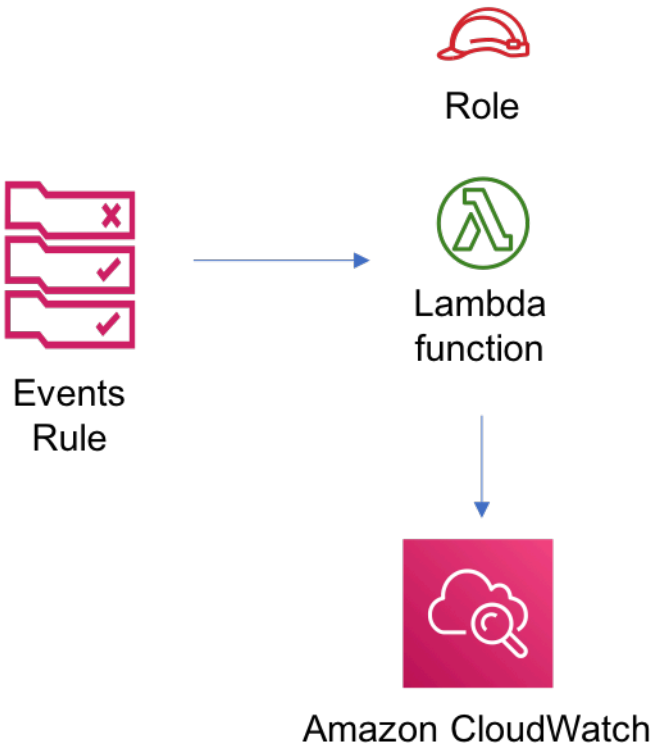
Reglas de Amazon CloudWatch Events

- Conceder permisos de privilegios mínimos a CloudWatch Events para activar la función Lambda

Función de AWS Lambda

- Configuración del rol de IAM de acceso limitado para la función Lambda
- Habilitar conexiones de reutilización con la función Keep-Alive para NodeJS Lambda
- Habilitar el X-Ray streo
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-events-rule-lambda

aws-events-rule-sns

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_events_rule_sns`
Mecanografía	`@aws-solutions-constructs/aws-events-rule-sns`
Java	`software.amazon.awsconstructs.services.eventsrulesns`

Overview

Este patrón implementa una regla de Amazon CloudWatch Events conectada a un tema de Amazon SNS.

Aquí hay una definición mínima de patrón implementable:

```
import { Duration } from '@aws-cdk/core';
import * as events from '@aws-cdk/aws-events';
import * as iam from '@aws-cdk/aws-iam';
import { EventsRuleToSnsProps, EventsRuleToSns } from "@aws-solutions-constructs/aws-
events-rule-sns";

const props: EventsRuleToSnsProps = {
    eventRuleProps: {
      schedule: events.Schedule.rate(Duration.minutes(5)),
    }
};

const constructStack = new EventsRuleToSns(this, 'test-construct', props);

// Grant yourself permissions to use the Customer Managed KMS Key
const policyStatement = new iam.PolicyStatement({
    actions: ["kms:Encrypt", "kms:Decrypt"],
    effect: iam.Effect.ALLOW,
    principals: [ new iam.AccountRootPrincipal() ],
    resources: [ "*" ]
});

constructStack.encryptionKey?.addToResourcePolicy(policyStatement);
```

Initializer

```
new EventsRuleToSNS(scope: Construct, id: string, props: EventsRuleToSNSProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`EventsRuleToSnsProps`

Patrón de construcción

Nombre	Tipo	Descripción
EventRuleProps	`events.RuleProps`	Propiedades proporcionadas por el usuario para anular las propiedades predeterm inadas de la regla CloudWatch Events.
¿ExistentetoPicobj?	`sns.Topic`	Instancia existente del objeto SNS Topic, proporcionando tanto esto como`topicProp s` provocará un error.
¿TopicProps?	`sns.TopicProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas del tema SNS. Se pasa por alto si un`existingTopicObj` se encuentra.
EnableEncryptionWi thCustomerManagedKey?	`boolean`	Si desea utilizar una clave de cifrado administrada por el cliente, ya sea administr ada por esta aplicación CDK o importada. Si se importa una clave de cifrado, debe especificarse en el campo`encryptionKey` para esta construcción.
¿encryptionKey?	`kms.Key`	Una clave de cifrado opcional existente que se utilizará en lugar de la clave de cifrado predeterminada.

Nombre	Tipo	Descripción
EncryptionKeyProps?	`kms.KeyProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la clave de cifrado.

Propiedades de patrón

Nombre	Tipo	Descripción
Regla de eventos	`events.Rule`	Devuelve una instancia de la regla Eventos creada por el patrón.
SNSTopic	`sns.Topic`	Devuelve una instancia del tema SNS creado por el patrón.
encryptionKey	`kms.Key`	Devuelve una instancia de la clave de cifrado creada por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Amazon CloudWatch Events, regla

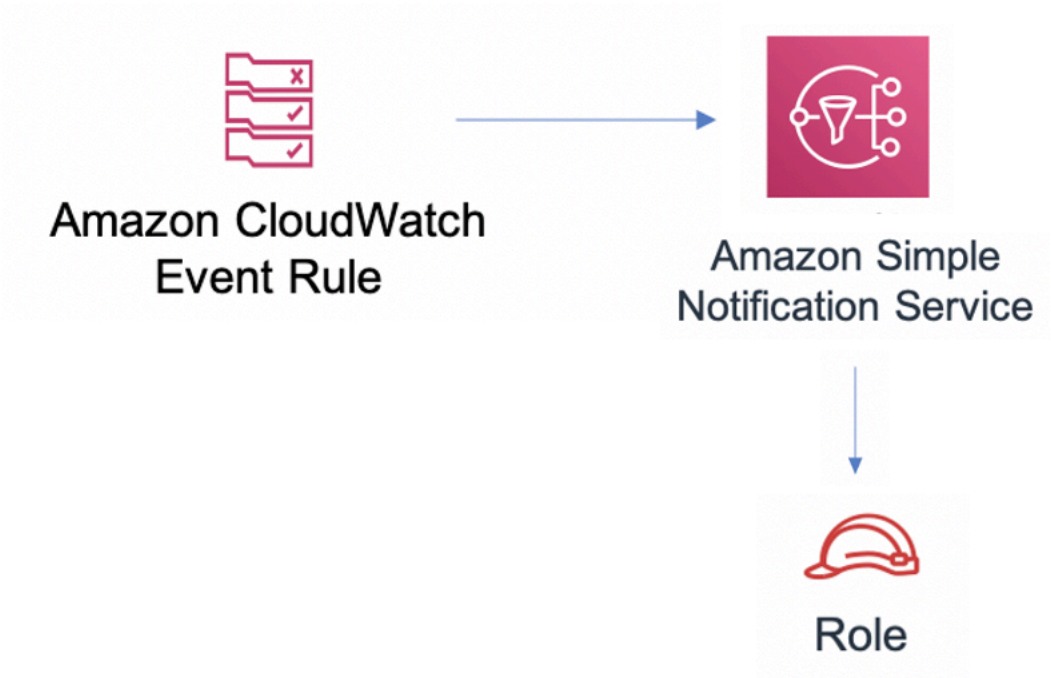
- Conceda permisos de privilegios mínimos a CloudWatch Events para publicar en el tema SNS.

Tema de Amazon SNS

- Configurar permisos de acceso de privilegios mínimos para el tema SNS.

- Habilite el cifrado en el lado de servidor para el tema SNS mediante la clave de AWS KMS administrada por el cliente.
- Aplicación del cifrado de los datos en tránsito.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-events-rule-sns

aws-events-rule-sqs

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersionado semánticoModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_events_rule_sqs`
Mecanografía	`@aws-solutions-constructs/aws-events-rule-sqs`
Java	`software.amazon.awsconstructs.services.eventsrulesqs`

Overview

Este patrón implementa una regla de Amazon CloudWatch Events conectada a una cola de Amazon SQS.

Aquí hay una definición mínima de patrón implementable:

```
import { Duration } from '@aws-cdk/core';
import * as events from '@aws-cdk/aws-events';
import * as iam from '@aws-cdk/aws-iam';
import { EventsRuleToSqsProps, EventsRuleToSqs } from "@aws-solutions-constructs/aws-events-rule-sqs";

const props: EventsRuleToSqsProps = {
    eventRuleProps: {
      schedule: events.Schedule.rate(Duration.minutes(5))
    }
```

```
};

const constructStack = new EventsRuleToSqs(this, 'test-construct', props);

// Grant yourself permissions to use the Customer Managed KMS Key
const policyStatement = new iam.PolicyStatement({
    actions: ["kms:Encrypt", "kms:Decrypt"],
    effect: iam.Effect.ALLOW,
    principals: [ new iam.AccountRootPrincipal() ],
    resources: [ "*" ]
});

constructStack.encryptionKey?.addToResourcePolicy(policyStatement);
```

Initializer

```
new EventsRuleToSqs(scope: Construct, id: string, props: EventsRuleToSqsProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`EventsRuleToSqsProps`

Patrón de construcción

Nombre	Tipo	Descripción
EventRuleProps	`events.RuleProps`	Propiedades proporcionadas por el usuario para anular las propiedades predeterm inadas de la regla CloudWatch Events.
ExistingQueueObj?	`sqs.Queue`	Una cola SQS opcional existente que se utilizará en

Nombre	Tipo	Descripción
		lugar de la cola predeterm inada. Proporcionando tanto esto como`queueProp s` provocará un error.
¿Props en cola?	`sqs.QueueProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas de la cola SQS. Se pasa por alto si un`existingQueueObj` se encuentra.
¿EnableQueuePurging?	`boolean`	Si se deben conceder permisos adicionales a la función Lambda, lo que le permite depurar la cola SQS. El valor predeterminado es `false`.
DeployDeadLetterQueue?	`boolean`	Si se debe crear una cola secundaria para utilizars e como cola de mensajes fallidos. El valor predeterm inado es `true`.
DeadletterQueueProps?	`sqs.QueueProps`	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para la cola de letras muertas. Solo se usa si el`deployDeadLetterQu eue` se establece en verdadero.

Nombre	Tipo	Descripción
¿maxReceiveCount?	`number`	El número de veces que un mensaje se puede quitar sin éxito antes de que se traslade a la cola de mensajes fallidos. El valor predeterminado es `15`.
EnableEncryptionWithCustomerManagedKey?	`boolean`	Si desea utilizar una clave de cifrado administrada por el cliente, ya sea administrada por esta aplicación CDK o importada. Si importa una clave de cifrado, debe especificarse en el campo`encryptionKey` para esta construcción.
encryptionKey	`kms.Key`	Una clave de cifrado opcional existente que se utilizará en lugar de la clave de cifrado predeterminada.
EncryptionKeyProps?	`kms.KeyProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la clave de cifrado.

Propiedades de patrón

Nombre	Tipo	Descripción
Regla de eventos	`events.Rule`	Devuelve una instancia de la regla Eventos creada por el patrón.

Nombre	Tipo	Descripción
SQSqueue	`sqs.Queue`	Devuelve una instancia de la cola SQS creada por el patrón.
encryptionKey	`kms.Key`	Devuelve una instancia de la clave de cifrado creada por el patrón.
DeadLetterQueue?	`sqs.Queue`	Devuelve una instancia de la cola de letras muertas creada por el patrón, si se implementa una.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

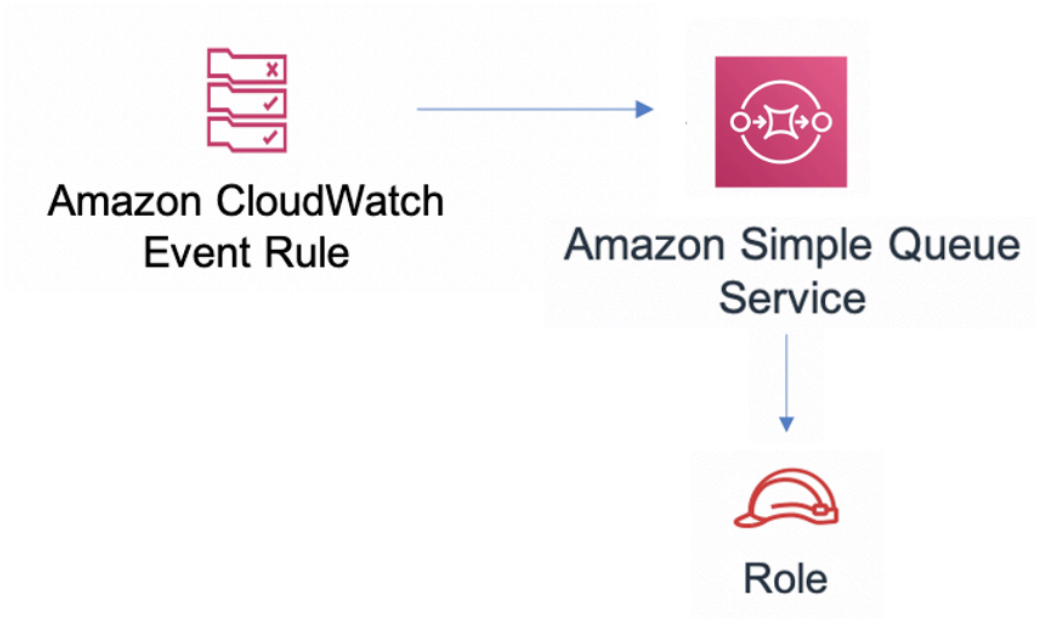
Amazon CloudWatch Events

- Conceda permisos de privilegios mínimos a CloudWatch Events para publicar en la cola de SQS.

Cola de Amazon SQS

- Implemente una cola de mensajes fallidos para la cola de origen.
- Active el cifrado del lado del servidor para la cola de origen mediante una clave de AWS KMS administrada por el cliente.
- Aplicar el cifrado de los datos en tránsito.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-events-rule-sqs

aws-events-rule-step-función

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones

semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_events_rule_step_function`
Mecanografía	`@aws-solutions-constructs/aws-events-rule-step-function`
Java	`software.amazon.awsconstructs.services.eventsrulestepfunction`

Overview

Este constructo de soluciones de AWS implementa una regla de eventos de AWS y una función de AWS Step.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { EventsRuleToStepFunction, EventsRuleToStepFunctionProps } from '@aws-solutions-constructs/aws-events-rule-step-function';

const startState = new stepfunctions.Pass(this, 'StartState');

const props: EventsRuleToStepFunctionProps = {
    stateMachineProps: {
      definition: startState
    },
    eventRuleProps: {
      schedule: events.Schedule.rate(Duration.minutes(5))
    }
```

```
};

new EventsRuleToStepFunction(this, 'test-events-rule-step-function-stack', props);
```

Initializer

```
new EventsRuleToStepFunction(scope: Construct, id: string, props:
 EventsRuleToStepFunctionProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`EventsRuleToStepFunctionProps`

Patrón de construcción

Nombre	Tipo	Descripción
StateMachineProps	`sfn.StateMachineProps`	Los apoyos proporcionados por el usuario opcionales para anular los apoyos predeterminados para SFN.State Machine
EventRuleProps	`events.RuleProps`	El usuario proporcionó EventRuleProps para anular los valores predeterminados
CreateCloudWatchalarms	`boolean`	Si desea crear alarmas recomendadas de CloudWatch.
LogGroupProps?	`logs.LogGroupProps`	Concede accesorios opcionales proporcionados por el

Nombre	Tipo	Descripción
		usuario para anular los accesorios predeterminados para el grupo de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
¿CloudWatchalarms?	`cloudwatch.Alarm[]`	Devuelve una lista de una o más alarmas de CloudWatch creadas por el patrón.
Regla de eventos	`events.Rule`	Devuelve una instancia de la regla Eventos creada por el patrón.
StateMachine	`sfn.StateMachine`	Devuelve una instancia de la máquina de estado creada por el patrón.
StateMachineLogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por el patrón para la máquina de estado.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

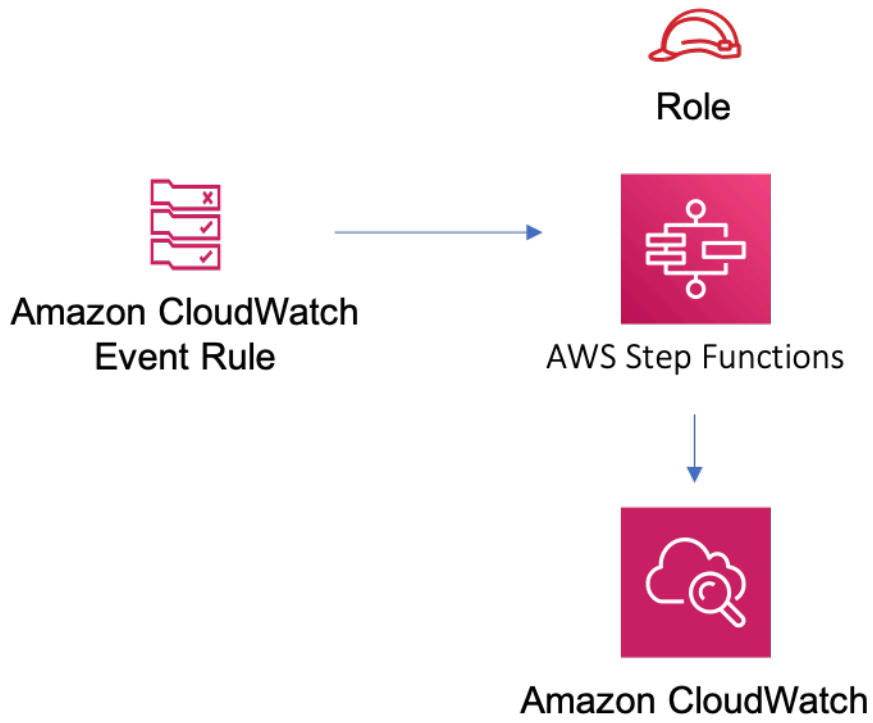
Amazon CloudWatch Events

- Conceder permisos de privilegios mínimos a CloudWatch Events para activar la función Lambda

AWS Step Functional

- Activar registro de CloudWatch para API Gateway
- Implementar las mejores prácticas de las alarmas CloudWatch para la función Step

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -solutions-constructs/aws-events-rule-step-function

aws-yot-kinesisfiremanse-s3

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones

semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_iot_kinesisfirehose_s3`
Mecanografía	`@aws-solutions-constructs/aws-iot-kinesisfirehose-s3`
Java	`software.amazon.awsconstructs.services.iotkinesisfirehoses3`

Overview

Este constructo de soluciones de AWS implementa una regla de tema de AWS IoT MQTT para enviar datos a una secuencia de entrega de Amazon Kinesis Data Firehose conectada a un bucket de Amazon S3.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { IotToKinesisFirehoseToS3Props, IotToKinesisFirehoseToS3 } from '@aws-solutions-constructs/aws-iot-kinesisfirehose-s3';

const props: IotToKinesisFirehoseToS3Props = {
    iotTopicRuleProps: {
        topicRulePayload: {
            ruleDisabled: false,
            description: "Persistent storage of connected vehicle telematics data",
            sql: "SELECT * FROM 'connectedcar/telemetry/#'",
            actions: []
        }
```

```
    }
};

new IotToKinesisFirehoseToS3(this, 'test-iot-firehose-s3', props);
```

Initializer

```
new IotToKinesisFirehoseToS3(scope: Construct, id: string, props:
 IotToKinesisFirehoseToS3Props);
```

Parámetros

- scope`Construct`
- id`string`
- props`IotToKinesisFirehoseToS3Props`

Patrón de construcción

Nombre	Tipo	Descripción
IottopicRuleProps	`iot.CfnTopicRuleProps`	El usuario proporcionó CFNTopICRuleProps para anular los valores predeterminados
¿KinesisFireHoseProps?	`kinesisfirehose.CfnDeliveryStreamProps`	El usuario ha proporcionado accesorios opcionales para anular los accesorios predeterminados para Kinesis Firehose Delivery Stream
¿ExistenteBuckeTobj?	`s3.Bucket`	Instancia existente del objeto S3 Bucket, proporcionando tanto esto como`bucketProps` provocará un error.

Nombre	Tipo	Descripción
¿BucketProps?	`s3.BucketProps`	El usuario proporcionó accesorios para anular los accesorios predeterm inados para el cubo S3. Si esto se proporciona, entonces también proporcio nabucketProps es un error.
LogGroupProps?	`logs.LogGroupProps`	Los accesorios opcionales proporcionados por el usuario para anular los accesorios predeterminados para el grupo de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
IotActionsRole	`iam.Role`	Devuelve una instancia de la función de IAM creada por el patrón de la regla IoT.
IotTopICRule	`iot.CfnTopicRule`	Devuelve una instancia de la regla de tema de IoT creada por el patrón.
KinesisFirehose	`kinesisfirehose.CfnDeliveryStream`	Devuelve una instancia de la secuencia de entrega de Kinesis Firehose creada por el patrón.
KinesisFireHoselogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por el patrón al que se envían los

Nombre	Tipo	Descripción
		registros de acceso de Kinesis Firehose.
KinesisFireHoserole	`iam.Role`	Devuelve una instancia del rol de IAM creado por el patrón para la secuencia de entrega de Kinesis Firehose.
¿S3 Bucket?	`s3.Bucket`	Devuelve una instancia del bucket S3 creado por el patrón.
¿S3LoggingBucket?	`s3.Bucket`	Devuelve una instancia del depósito de registro creado por el patrón para el bucket S3.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Regla de Amazon IoT

- Configurar el rol de IAM de acceso mínimo con privilegios para Amazon IoT

Amazon Kinesis Firehose

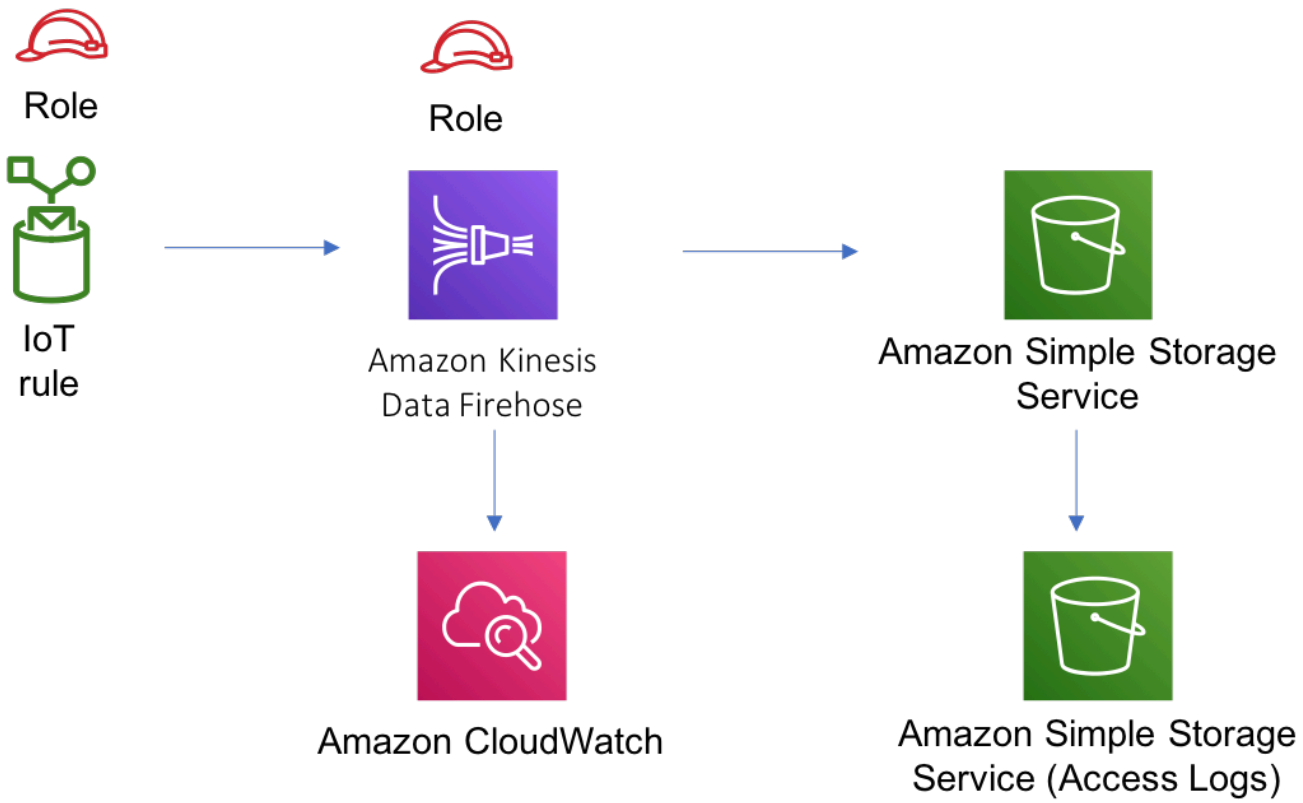
- Habilitar el registro de CloudWatch para Kinesis Firehose
- Configurar el rol de IAM de acceso mínimo con privilegios para Amazon Kinesis Firehose

Bucket de Amazon S3

- Configurar el registro de acceso para S3 Bucket

- Habilitar el cifrado del lado del servidor para S3 Bucket utilizando la clave KMS administrada por AWS
- Activar el control de versiones para S3 Bucket
- No permitir el acceso público para S3 Bucket
- Conservar el depósito de S3 al eliminar la pila de CloudFormation
- Aplica una regla de ciclo de vida para mover versiones de objetos no actuales al almacenamiento de Glacier después de 90 días

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-yot-kinesisfirehose-s3

aws-yot-lambda

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersionado semánticomodelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_iot_lambda`
Mecanografía	`@aws-solutions-constructs/aws-iot-lambda`
Java	`software.amazon.awsconstructs.services.iotlambda`

Overview

Este patrón de AWS Solutions Constructs implementa una regla de tema de AWS IoT MQTT y un patrón de función de AWS Lambda.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { IotToLambdaProps, IotToLambda } from '@aws-solutions-constructs/aws-iot-
lambda';

const props: IotToLambdaProps = {
    lambdaFunctionProps: {
```

```
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    },
    iotTopicRuleProps: {
        topicRulePayload: {
            ruleDisabled: false,
            description: "Processing of DTC messages from the AWS Connected Vehicle
 Solution.",
            sql: "SELECT * FROM 'connectedcar/dtc/#'",
            actions: []
        }
    }
};

new IotToLambda(this, 'test-iot-lambda-integration', props);
```

Initializer

```
new IotToLambda(scope: Construct, id: string, props: IotToLambdaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`IotToLambdaProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFun`

Nombre	Tipo	Descripción
		`ctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si `unexistingLambdaObj` se encuentra a su disposición.
¿IottopicruleProps?	`iot.CfnTopicRulePr ops`	El usuario proporcionó CFNTopICRuleProps para anular los valores predeterm inados

Propiedades de patrón

Nombre	Tipo	Descripción
IotTopICRule	`iot.CfnTopicRule`	Devuelve una instancia de la regla de tema de IoT creada por el patrón.
LambdaFinction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

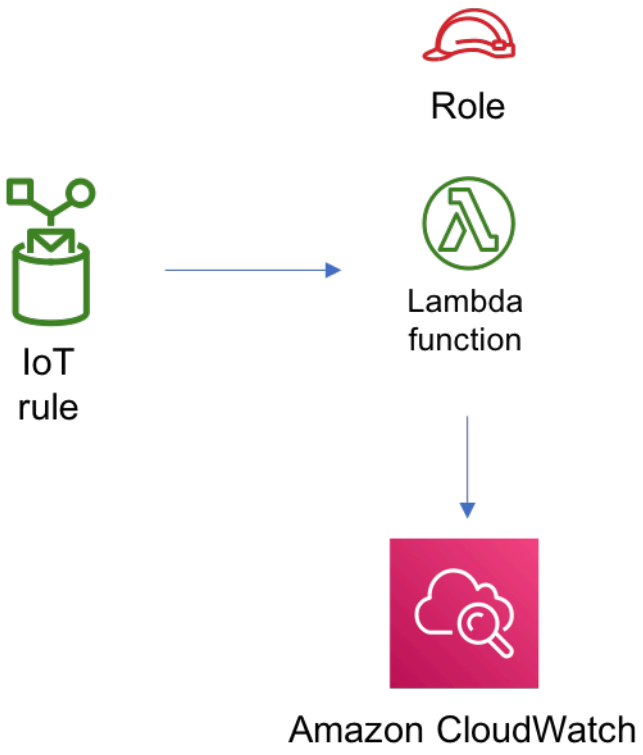
Regla de Amazon IoT

- Configurar el rol de IAM de acceso mínimo con privilegios para Amazon IoT.

Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Active el rastreo de X-Ray.
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-iot-lambda

aws-iot-lambda-dynamodb

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_iot_lambda_dynamodb`
Mecanografía	`@aws-solutions-constructs/aws-iot-lambda-dynamodb`
Java	`software.amazon.awsconstructs.services.iotlambdadynamodb`

Overview

Este patrón de AWS Solutions Constructs implementa una regla temática de AWS IoT, una función de AWS Lambda y una tabla de Amazon DynamoDB con los permisos menos privilegiados.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { IotToLambdaToDynamoDBProps,  IotToLambdaToDynamoDB } from '@aws-solutions-
constructs/aws-iot-lambda-dynamodb';

const props: IotToLambdaToDynamoDBProps = {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    },
    iotTopicRuleProps: {
        topicRulePayload: {
            ruleDisabled: false,
            description: "Processing of DTC messages from the AWS Connected Vehicle
 Solution.",
            sql: "SELECT * FROM 'connectedcar/dtc/#'",
            actions: []
        }
    }
};

new IotToLambdaToDynamoDB(this, 'test-iot-lambda-dynamodb-stack', props);
```

Initializer

```
new IotToLambdaToDynamoDB(scope: Construct, id: string, props:
 IotToLambdaToDynamoDBProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`IotToLambdaToDynamoDBProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
LambdaFontionProps	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si `unexistingLambdaObj` se encuentra a su disposición.
IottopicRuleProps	`iot.CfnTopicRuleProps`	Proporcionadas por el usuario los accesorios predeterminados
¿DynamotableProps?	`dynamodb.TableProps`	Los accesorios proporcionados por el usuario opcionales para anular los accesorios predeterminados para la tabla de DynamoDB
¿Permisos de tabla?	`string`	Permisos de tabla opcionales que se concederán a la función Lambda. Puede que se especifique una de las siguientes opciones: `All`,`Read`,`ReadWrite` , o bien`Write`.

Propiedades de patrón

Nombre	Tipo	Descripción
DynamoTable	`dynamodb.Table`	Devuelve una instancia de la tabla DynamoDB creada por el patrón.
IotTopICRule	`iot.CfnTopicRule`	Devuelve una instancia de la regla de tema de IoT creada por el patrón.
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Regla de Amazon IoT

- Configurar el rol de IAM de acceso mínimo con privilegios para Amazon IoT.

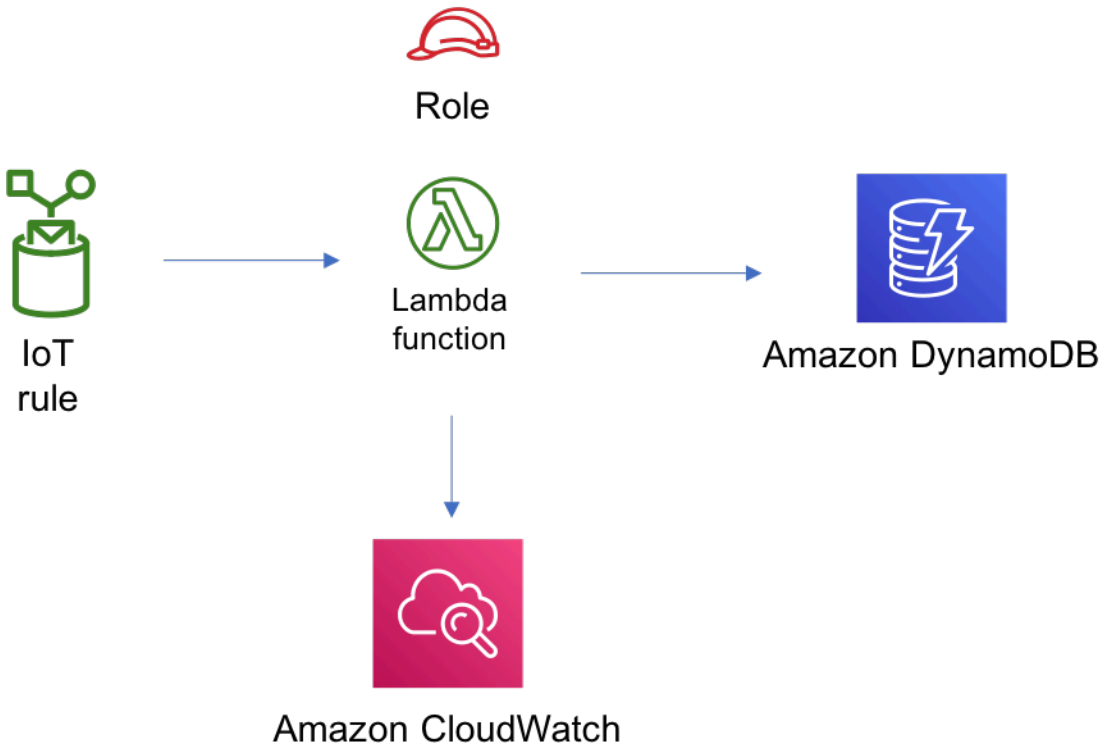
Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Active el rastreo de X-Ray.
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Amazon DynamoDB Tabla

- Establezca el modo de facturación para la tabla de DynamoDB en Bajo demanda (pago por solicitud).
- Habilite el cifrado del lado del servidor para DynamoDB Table mediante la clave KMS administrada por AWS.
- Crea una clave de partición llamada 'id' para DynamoDB Table.
- Conservar la tabla al eliminar la pila CloudFormation.
- Habilite las copias de seguridad continuas y la recuperación a un momento dado.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-iot-lambda-dynamodb

aws-kinesisfiremanse-s3

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersionado semánticoModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws-kinesis-firehose-s3`
TypeScript	`@aws-solutions-constructs/aws-kinesisfirehose-s3`
Java	`software.amazon.awsconstructs.services.kinesisfirehoses3`

Overview

Este constructo de soluciones de AWS implementa un flujo de entrega de Amazon Kinesis Data Firehose conectado a un bucket de Amazon S3.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { KinesisFirehoseToS3 } from '@aws-solutions-constructs/aws-kinesisfirehose-s3';

new KinesisFirehoseToS3(this, 'test-firehose-s3', {});
```

Initializer

```
new KinesisFirehoseToS3(scope: Construct, id: string, props: KinesisFirehoseToS3Props);
```

Parámetros

- scope`Construct`
- id`string`
- props`KinesisFirehoseToS3Props`

Patrón de construcción

Nombre	Tipo	Descripción
¿BucketProps?	`s3.BucketProps`	El usuario ha proporcionado accesorios opcionales para anular los accesorios predeterminados para el cubo S3.
¿ExistenteBuckeTobj?	`s3.IBucket`	Instancia existente opcional de S3 Bucket. Si se proporciona esto, entonces también proporcionar `bucketProps` es un error.
ExistingLogingBuckeTobj?	`s3.IBucket`	Instancia existente opcional de registro de S3 Bucket para el S3 Bucket creado por el patrón.
¿KinesisFireHoseProps?	`kinesisfirehose.CfnDeliveryStreamProps` | `any`	El usuario ha proporcionado accesorios opcionales para anular los accesorios

Nombre	Tipo	Descripción
		predeterminados para Kinesis Firehose Delivery Stream.
LogGroupProps?	logs.LogGroupProps	El usuario ha proporcionado accesorios opcionales para anular los accesorios predeterminados para el LogGroup de CloudWatchLogLogs.

Propiedades de patrón

Nombre	Tipo	Descripción
KinesisFirehose	kinesisfirehose.CfnDeliveryStream	Devuelve una instancia de KinesisFireHose.cfnDeliveryStream creada por la construcción.
KinesisFireHoselogGroup	logs.LogGroup	Devuelve una instancia del logs.logGroup creado por la construcción para la secuencia de entrega de Kinesis Data Firehose.
KinesisFireHoserole	iam.Role	Devuelve una instancia del IAM.Role creado por la construcción para el flujo de entrega de Kinesis Data Firehose.
¿Bucket de S3?	s3.Bucket	Devuelve una instancia de S3.Bucket creada por la construcción.

Nombre	Tipo	Descripción
¿S3LoggingBucket?	`s3.Bucket`	Devuelve una instancia de S3.Bucket creada por la construcción como depósito de registro para el depósito principal.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

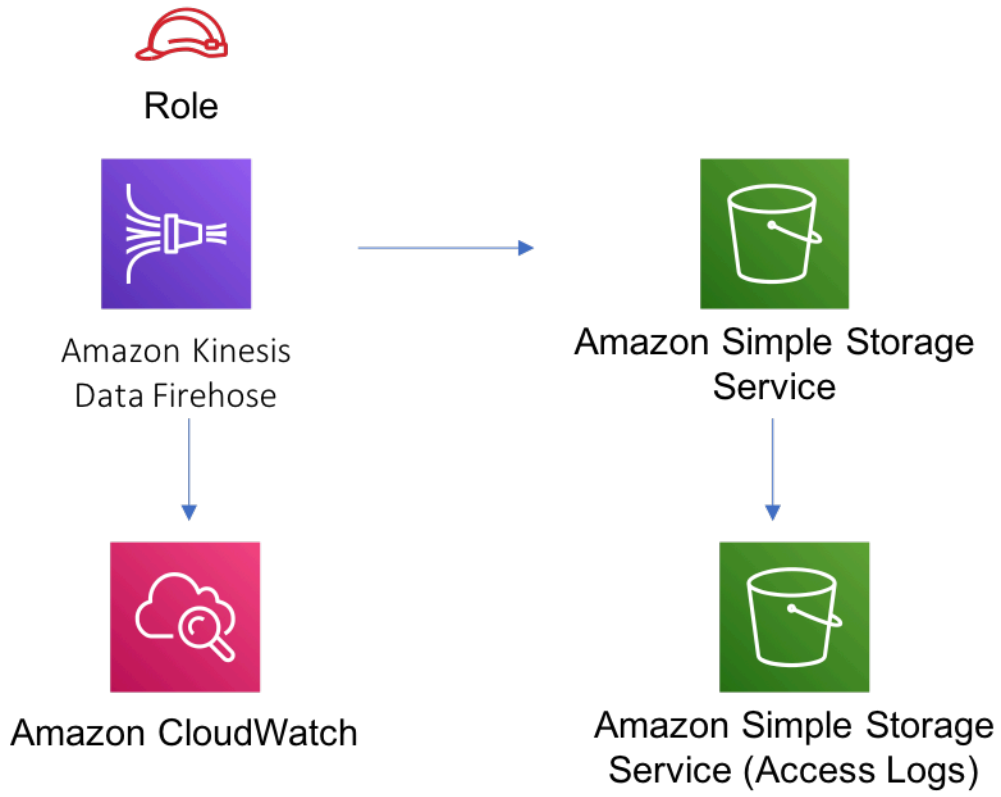
Amazon Kinesis Firehose

- Habilitar el registro de CloudWatch para Kinesis Firehose
- Configurar el rol de IAM de acceso mínimo con privilegios para Amazon Kinesis Firehose

Bucket de Amazon S3

- Configurar el registro de acceso para S3 Bucket
- Habilitar el cifrado del lado del servidor para S3 Bucket utilizando la clave KMS administrada por AWS
- Activar el control de versiones para S3 Bucket
- No permitir el acceso público para S3 Bucket
- Conservar el depósito de S3 al eliminar la pila de CloudFormation
- Aplicación del cifrado de los datos en tránsito
- Aplica una regla de ciclo de vida para mover versiones de objetos no actuales al almacenamiento de Glacier después de 90 días

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-kinesisfirehose-s3

aws-kinesisfiremanse-s3-y-kinesisanalytics

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Éstos no están sujetos a laVersionado semánticoModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_kinesisfirehose_s3_and_kinesisanalytics`
Mecanografía	`@aws-solutions-constructs/aws-kinesisfirehose-s3-and-kinesisanalytics`
Java	`software.amazon.awsconstructs.services.kinesisfirehose s3kinesisanalytics`

Overview

Este constructo de soluciones de AWS implementa una transmisión de entrega de Amazon Kinesis Firehose conectada a un bucket de Amazon S3 y una aplicación de Amazon Kinesis Analytics.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { KinesisFirehoseToAnalyticsAndS3 } from '@aws-solutions-constructs/aws-
kinesisfirehose-s3-and-kinesisanalytics';

new KinesisFirehoseToAnalyticsAndS3(this, 'FirehoseToS3AndAnalyticsPattern', {
    kinesisAnalyticsProps: {
        inputs: [{
            inputSchema: {
                recordColumns: [{
                    name: 'ticker_symbol',
                    sqlType: 'VARCHAR(4)',
                    mapping: '$.ticker_symbol'
                }, {
                    name: 'sector',
                    sqlType: 'VARCHAR(16)',
                    mapping: '$.sector'
```

```
                }, {
                    name: 'change',
                    sqlType: 'REAL',
                    mapping: '$.change'
                }, {
                    name: 'price',
                    sqlType: 'REAL',
                    mapping: '$.price'
                }],
                recordFormat: {
                    recordFormatType: 'JSON'
                },
                recordEncoding: 'UTF-8'
            },
            namePrefix: 'SOURCE_SQL_STREAM'
        }]
    }
});
```

Initializer

```
new KinesisFirehoseToAnalyticsAndS3(scope: Construct, id: string, props:
 KinesisFirehoseToAnalyticsAndS3Props);
```

Parámetros

- scope`Construct`
- id`string`
- props`KinesisFirehoseToAnalyticsAndS3Props`

Patrón de construcción

Nombre	Tipo	Descripción
¿KinesisFireHoseProps?	`kinesisFirehose.CfnDeliveryStreamProps`	Props opcionales proporcionados por el usuario para anular los accesorios

Nombre	Tipo	Descripción
		predeterminados para la transmisión de entrega de Kinesis Firehose.
KinesisanalyticsProps?	`kinesisAnalytics.CfnApplicationProps`	Props opcionales proporcionados por el usuario para anular los accesorios predeterminados para la aplicación Kinesis Analytics.
¿ExistenteBuckeTobj?	`s3.IBucket`	Instancia existente del objeto S3 Bucket. Si se proporciona esto, entonces también proporcionabucketProps es un error. |
¿BucketProps?	`s3.BucketProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas del depósito. Se pasa por alto si unexistingBucketObj se encuentra.
LogGroupProps?	`logs.LogGroupProps`	Proporcionadas por el usuario opcionales para anular los accesorios predeterminados para el grupo de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
Kinesisanalytics	kinesisAnalytics.CfnApplication	Devuelve una instancia de la aplicación Kinesis Analytics creada por el patrón.
KinesisFirehose	kinesisfirehose.CfnDeliveryStream	Devuelve una instancia de la secuencia de entrega de Kinesis Firehose creada por el patrón.
KinesisFireHoselogGroup	logs.LogGroup	Devuelve una instancia del grupo de registros creado por el patrón al que se envían los registros de acceso de Kinesis Firehose.
KinesisFireHoserole	iam.Role	Devuelve una instancia del rol de IAM creado por el patrón para la secuencia de entrega de Kinesis Firehose.
¿S3 Bucket?	s3.Bucket	Devuelve una instancia del bucket S3 creado por el patrón.
¿S3LoggingBucket?	s3.Bucket	Devuelve una instancia del depósito de registro creado por el patrón para el bucket S3.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Amazon Kinesis Firehose

- Habilitar el registro de CloudWatch para Kinesis Firehose
- Configurar el rol de IAM de acceso mínimo con privilegios para Amazon Kinesis Firehose

Bucket de Amazon S3

- Configurar el registro de acceso para S3 Bucket
- Habilitar el cifrado del lado del servidor para S3 Bucket utilizando la clave KMS administrada por AWS
- Activar el control de versiones para S3 Bucket
- No permitir el acceso público para S3 Bucket
- Conservar el depósito de S3 al eliminar la pila de CloudFormation
- Aplicación del cifrado de los datos en tránsito
- Aplica una regla de ciclo de vida para mover versiones de objetos no actuales al almacenamiento de Glacier después de 90 días

Amazon Kinesis Data Analytics

- Configurar el rol de IAM de acceso mínimo con privilegios para Amazon Kinesis Analytics

Architecture

GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-kinesisfiremanse-s3-y-kinesisanalytics

aws-kinesisstreams-gluejob

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersión semánticaModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_kinesis_streams_gluejob`
Mecanografía	`@aws-solutions-constructs/aws-kinesisstreams-gluejob`
Java	`software.amazon.awsconstructs.services.kinesisstreamsgluejob`

Overview

Este constructo de soluciones de AWS implementa un flujo de datos de Amazon Kinesis y configura un Job de AWS Glue para realizar una transformación ETL personalizada con los recursos y propiedades adecuados para la interacción y la seguridad. También crea un depósito de Amazon S3 donde se puede cargar el script de Python para el Job de AWS Glue.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import * as glue from '@aws-cdk/aws-glue';
import * as s3assets from '@aws-cdk/aws-s3-assets';
import { KinesisstreamsToGluejob } from '@aws-solutions-constructs/aws-kinesisstreams-
gluejob';

const fieldSchema: glue.CfnTable.ColumnProperty[] = [
    {
        name: 'id',
        type: 'int',
        comment: 'Identifier for the record',
    },
    {
        name: 'name',
        type: 'string',
        comment: 'Name for the record',
    },
```

```
    {
        name: 'address',
        type: 'string',
        comment: 'Address for the record',
    },
    {
        name: 'value',
        type: 'int',
        comment: 'Value for the record',
    },
];

const customEtlJob = new KinesisstreamsToGluejob(this, 'CustomETL', {
    glueJobProps: {
        command: {
            name: 'gluestreaming',
            pythonVersion: '3',
            scriptLocation: new s3assets.Asset(this, 'ScriptLocation', {
                path: `${__dirname}/../etl/transform.py`,
            }).s3ObjectUrl,
        },
    },
    fieldSchema: fieldSchema,
});
```

Initializer

```
new KinesisstreamsToGluejob(scope: Construct, id: string, props:
 KinesisstreamsToGluejobProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`KinesisstreamsToGluejobProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿KinesisStreamProps?	`kinesis.StreamProps`	Props opcionales proporcionados por el usuario para anular los accesorios predeterminados para Amazon Kinesis Data Stream.
¿ExistenteStreamObj?	`kinesis.Stream`	Instancia existente de Kinesis Stream, proporcionando tanto esto como`kinesisStreamProps` provocará un error.
¿Gluejobprops?	`cfnJob.CfnJobProps`	Los accesorios proporcionados por el usuario para anular los accesorios predeterminados para el trabajo de AWS Glue.
¿Existe GlueJob?	`cfnJob.CfnJob`	Instancia existente de AWS Glue Job, proporcionando tanto esto como`glueJobProps` provocará un error.
¿La base de datos existente?	`CfnDatabase`	Base de datos existente de AWS Glue que se utilizará con esta construcción. Si esto está configurado, entonces`databaseProps` Se pasa por alto.
¿Props de Base de Datos?	`CfnDatabaseProps`	Los accesorios proporcionados por el usuario para anular los accesorios predeterminados utilizados

Nombre	Tipo	Descripción
		para crear la base de datos de AWS Glue.
¿ExistingTable?	`CfnTable`	Instancia existente de la tabla AWS Glue. Si esto está configurado, entonces `tableProps` y `fieldSchema` Se ignoran.
¿TableProps?	`CfnTableProps`	Los apoyos proporcionados por el usuario para anular los accesorios predeterminados utilizados para crear una tabla de AWS Glue.
¿Esquema de camposEsquema?	`CfnTable.ColumnProperty[]`	Estructura de esquema proporcionada por el usuario para crear una tabla de AWS Glue.
OutputDataStore?	`SinkDataStoreProps`	Los accesorios proporcionados por el usuario para un depósito de Amazon S3 que almacena los resultados del trabajo de AWS Glue. Actualmente solo admite Amazon S3 como tipo de almacén de datos de salida.

SinkDataStoreProps

Nombre	Tipo	Descripción
Existentes3OutputBucket?	`Bucket`	Instancia existente de bucket de S3 donde se deben escribir los datos. Proporcionando tanto esto como`outputBucketProps` provocará un error.
OutputBucketProps	`BucketProps`	Propiedades de depósito proporcionadas por el usuario para crear el depósito de Amazon S3 utilizado para almacenar la salida del trabajo de AWS Glue.
DataStoreType	`SinkStoreType`	Tipo de almacén de datos del receptor.

SinkStoreType

Enumeración de tipos de data store que podrían incluir S3, DynamoDB, DocumentDB, RDS o Redshift. La implementación actual de construcciones solo admite S3, pero puede agregar otros tipos de salida en el futuro.

Nombre	Tipo	Descripción
S3	`string`	Tipo de almacenamiento S3

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Amazon Kinesis

- Configure el rol de IAM de acceso mínimo con privilegios para Amazon Kinesis Data Stream.
- Habilite el cifrado del lado del servidor para Amazon Kinesis Stream mediante una clave KMS administrada de AWS.
- Implemente las mejores prácticas de las alarmas de Amazon CloudWatch para Amazon Kinesis Stream.

Glue Job

- Cree una configuración de seguridad de AWS Glue que configure el cifrado para CloudWatch, Job Marcadores y S3. CloudWatch y los marcadores de Job se cifran mediante la clave KMS administrada de AWS creada para AWS Glue Service. El bucket S3 se configura con el modo de cifrado SSE-S3.
- Configure directivas de función de servicio que permitan a AWS Glue leer un flujo de Amazon Kinesis Data Streams.

Base Glue datos

- Cree una base de datos AWS Glue. Se agregará una tabla de AWS Glue a la base de datos. En esta tabla se define el esquema de los registros almacenados en búfer en Amazon Kinesis Data Stream.

Glue e

- Cree una tabla de AWS Glue. La definición del esquema de tabla se basa en la estructura JSON de los registros almacenados en búfer en Amazon Kinesis Data Stream.

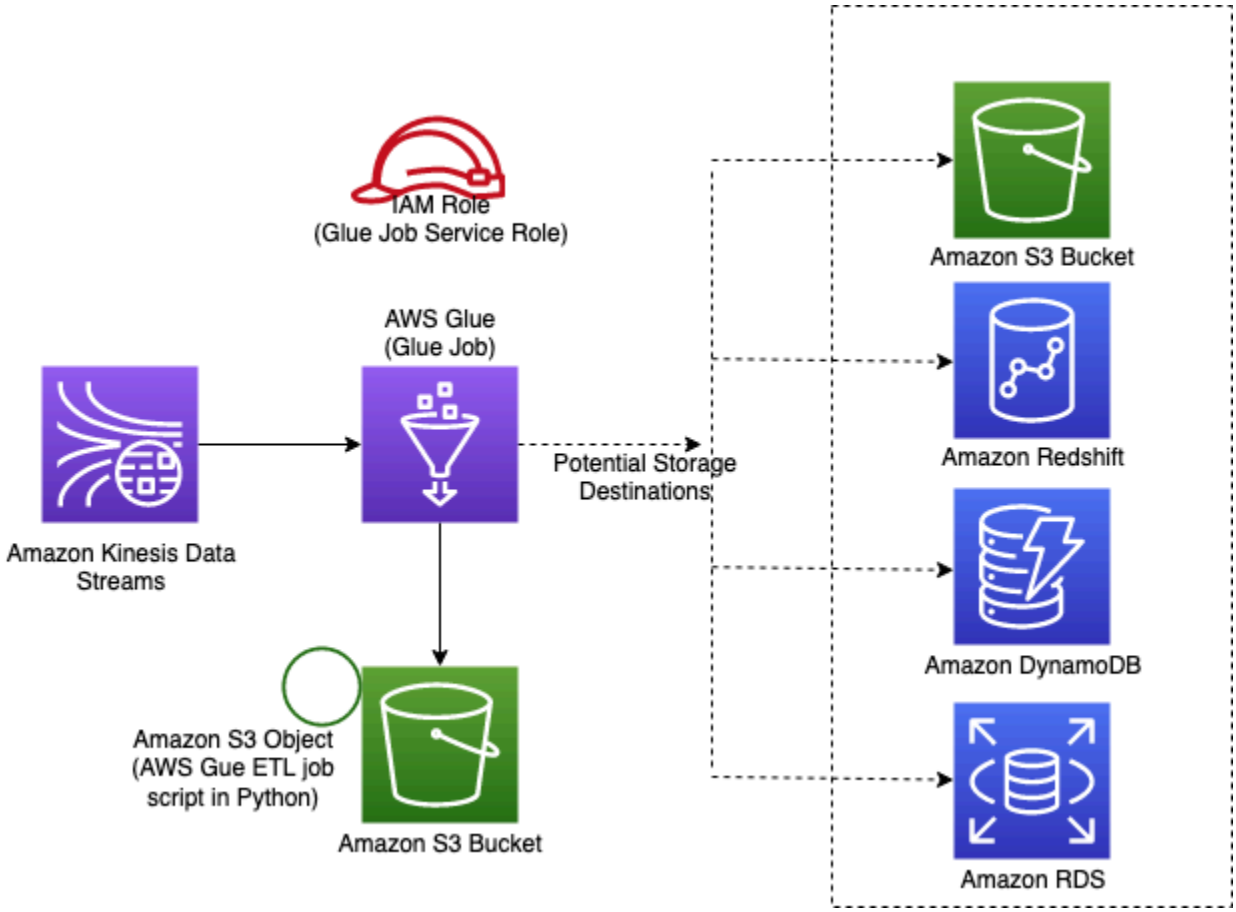
Función de IAM

- Función de ejecución de trabajos que tiene privilegios de 1) leer el script ETL desde la ubicación del depósito de Amazon S3, 2) leer registros del flujo de datos de Amazon Kinesis y 3) ejecutar el trabajo de Amazon Glue.

Cucharón S3 de salida

- Un bucket de Amazon S3 donde se almacenan los resultados de la transformación de ETL. Este depósito se pasará como argumento al trabajo de AWS Glue creado para que pueda utilizarse en el script ETL para escribir datos en él.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	[@aws -soluciones-constructs/aws-kinesisstreams-gluejob]

aws-kinesisstreams-kinesisfiremanse-s3

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_kinesisstreams_kinesisfirehose_s3`
Mecanografía	`@aws-solutions-constructs/aws-kinesis-streams-kinesis-firehose-s3`
Java	`software.amazon.awsconstructs.services.kinesisstreamskinesisfirehoses3`

Overview

Este constructo de soluciones de AWS implementa una secuencia de datos de Amazon Kinesis (KDS) conectada a una secuencia de entrega de Amazon Kinesis Data Firehose (KDF) conectada a un bucket de Amazon S3.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { KinesisStreamsToKinesisFirehoseToS3 } from '@aws-solutions-constructs/aws-kinesisstreams-kinesisfirehose-s3';
```

```
new KinesisStreamsToKinesisFirehoseToS3(this, 'test-stream-firehose-s3', {});
```

Initializer

```
new KinesisStreamsToKinesisFirehoseToS3(scope: Construct, id: string, props:
 KinesisStreams...ToS3Props);
```

Parámetros

- scope`Construct`
- id`string`
- props`KinesisStreams...ToS3Props`

Patrón de construcción

Nombre	Tipo	Descripción
¿BucketProps?	`s3.BucketProps`	El usuario ha proporcio nado accesorios opcionale s para anular los accesorios predeterminados para el cubo S3.
¿CreaCloudWatchalarms?	`boolean`	Opcional si se deben crear alarmas recomendadas de CloudWatch.
¿ExistenteBuckeTobj?	`s3.IBucket`	Instancia existente opcional del objeto S3 Bucket. Si se proporciona esto, entonces también proporcio nabucketProps es un error.
ExistingLogingBuckeTobj?	`s3.IBucket`	Instancia existente opcional de registro del objeto Bucket S3

Nombre	Tipo	Descripción
		para el cubo S3 creado por el patrón.
¿ExistenteStreamObj?	`kinesis.Stream`	Instancia existente de Kinesis Stream, proporcionando tanto esto como `kinesisStreamProps` provocará un error.
¿KinesisFireHoseProps?	`aws-kinesisfirehose.CfnDeliveryStreamProps` | `any`	El usuario ha proporcionado accesorios opcionales para anular los accesorios predeterminados para Kinesis Firehose Delivery Stream.
¿KinesisStreamProps?	`kinesis.StreamProps`	El usuario ha proporcionado accesorios opcionales para anular los accesorios predeterminados para la transmisión de Kinesis.
LogGroupProps?	`logs.LogGroupProps`	El usuario ha proporcionado accesorios opcionales para anular los accesorios predeterminados para el grupo de registro de CloudWatchLogs.

Propiedades de patrón

Nombre	Tipo	Descripción
¿Cloudwatchalarms?	`cloudwatch.Alarm[]`	Devuelve una lista de instancias de CloudWatc

Nombre	Tipo	Descripción
		h.Alarm creadas por el componente fijo.
KinesisFirehose	`kinesisfirehose.CfnDeliveryStream`	Devuelve una instancia de KinesisFireHose.cfnDeliveryStream creada por la construcción.
KinesisFireHoselogGroup	`logs.LogGroup`	Devuelve una instancia del logs.logGroup creado por la construcción para la secuencia de entrega de Kinesis Data Firehose.
KinesisFireHoserole	`iam.Role`	Devuelve una instancia del IAM.Role creado por la construcción para el flujo de entrega de Kinesis Data Firehose.
KinesisStreamRole	`iam.Role`	Devuelve una instancia del IAM.Role creado por la construcción para la secuencia Kinesis.
¿Bucket S3?	`s3.Bucket`	Devuelve una instancia de S3.Bucket creada por la construcción.
¿S3LoggingBucket?	`s3.Bucket`	Devuelve una instancia de S3.Bucket creada por la construcción como depósito de registro para el depósito principal.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Amazon Kinesis Stream

- Configurar el rol de IAM de acceso mínimo con privilegios para Kinesis Stream
- Habilitar el cifrado del lado del servidor para Kinesis Stream mediante la clave KMS administrada de AWS
- Implementar las mejores prácticas de las alarmas de CloudWatch para Kinesis Stream

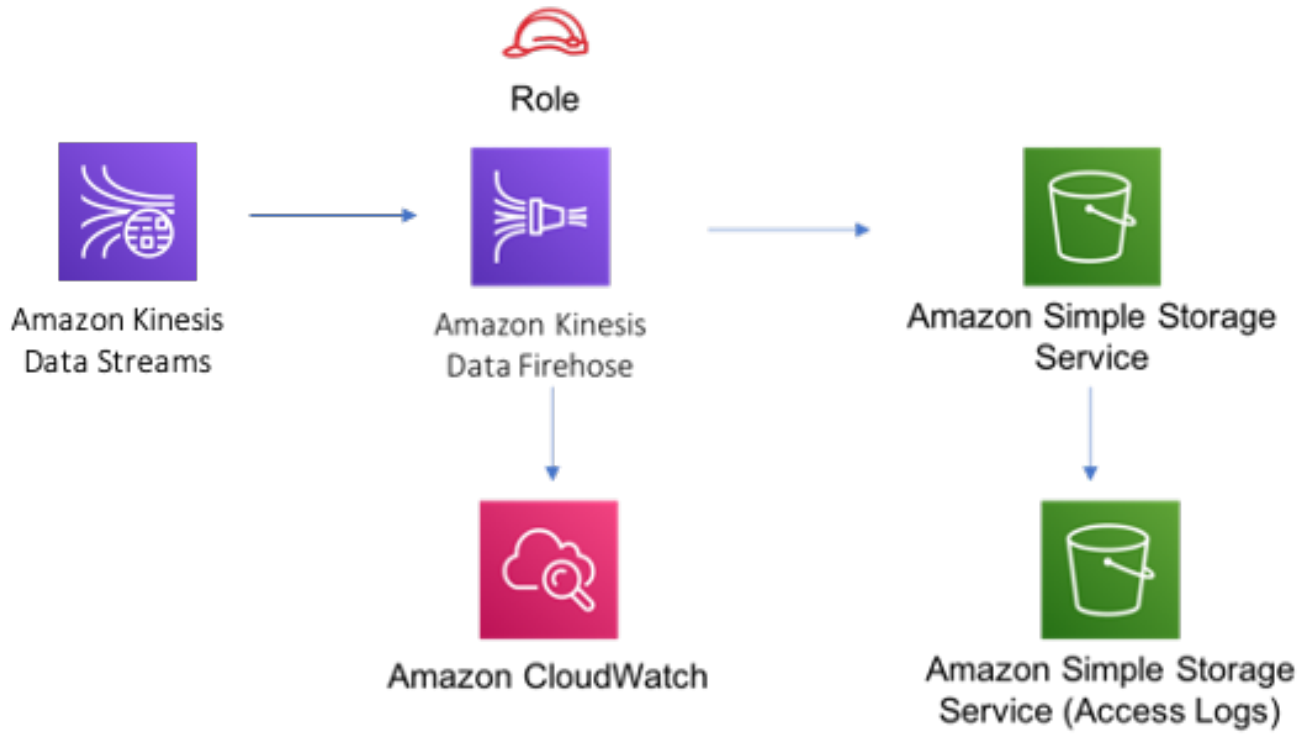
Amazon Kinesis Firehose

- Habilitar el registro de CloudWatch para Kinesis Firehose
- Configurar el rol de IAM de acceso mínimo con privilegios para Amazon Kinesis Firehose

Bucket de Amazon S3

- Configurar el registro de acceso para bucket de S3
- Habilitar el cifrado del lado del servidor para el bucket S3 mediante la clave KMS administrada por AWS
- Aplicación del cifrado de los datos en tránsito
- Habilitar el control de versiones de bucket
- No permitir el acceso público para el bucket S3
- Conservar el depósito de S3 al eliminar la pila de CloudFormation
- Aplicar regla de ciclo de vida para mover versiones de objetos no actuales al almacenamiento de Glacier después de 90 días

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-kinesisstreams-kinesisfirehose-s3

aws-kinesisstreams-lambda

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticas. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws-kinesis-streams-lambda`
Mecanografía	`@aws-solutions-constructs/aws-kinesisstreams-lambda`
Java	`software.amazon.awsconstructs.services.kinesisstreamslambda`

Overview

Este constructo de soluciones de AWS implementa una función Kinesis Stream y Lambda con los recursos y propiedades adecuados para la interacción y la seguridad.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { KinesisStreamsToLambda } from '@aws-solutions-constructs/aws-kinesisstreams-
lambda';

new KinesisStreamsToLambda(this, 'KinesisToLambdaPattern', {
    kinesisEventSourceProps: {
        startingPosition: lambda.StartingPosition.TRIM_HORIZON,
        batchSize: 1
    },
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    }
});
```

Initializer

```
new KinesisStreamsToLambda(scope: Construct, id: string, props:
 KinesisStreamsToLambdaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`KinesisStreamsToLambdaProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si se`existingLambdaObj` se encuentra.
¿KinesisStreamProps?	`kinesis.StreamProps`	Props opcionales proporcionados por el usuario para anular los apoyos predeterminados para la transmisión de Kinesis.

Nombre	Tipo	Descripción
¿ExistenteStreamObj?	`kinesis.Stream`	Instancia existente de Kinesis Stream, proporcionando tanto esto como `kinesisStreamProps` provocará un error.
¿KinesisEventSourceProps?	`aws-lambda-event-sources.KinesisEventSourceProps`	Props opcionales proporcionados por el usuario para anular los apoyos predeterminados para la asignación de origen de eventos Lambda.
CreateCloudWatchalarms	`boolean`	Si desea crear alarmas recomendadas de CloudWatch.

Propiedades de patrón

Nombre	Tipo	Descripción
Flujos de	`kinesis.Stream`	Devuelve una instancia de la secuencia Kinesis creada por el patrón.
Lambdafunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.
KinesisStreamRole	`iam.Role`	Devuelve una instancia del rol de IAM creado por el patrón para la secuencia de Kinesis.
¿Cloudwatchalarms?	`cloudwatch.Alarm[]`	Devuelve una lista de una o más alarmas de CloudWatch creadas por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

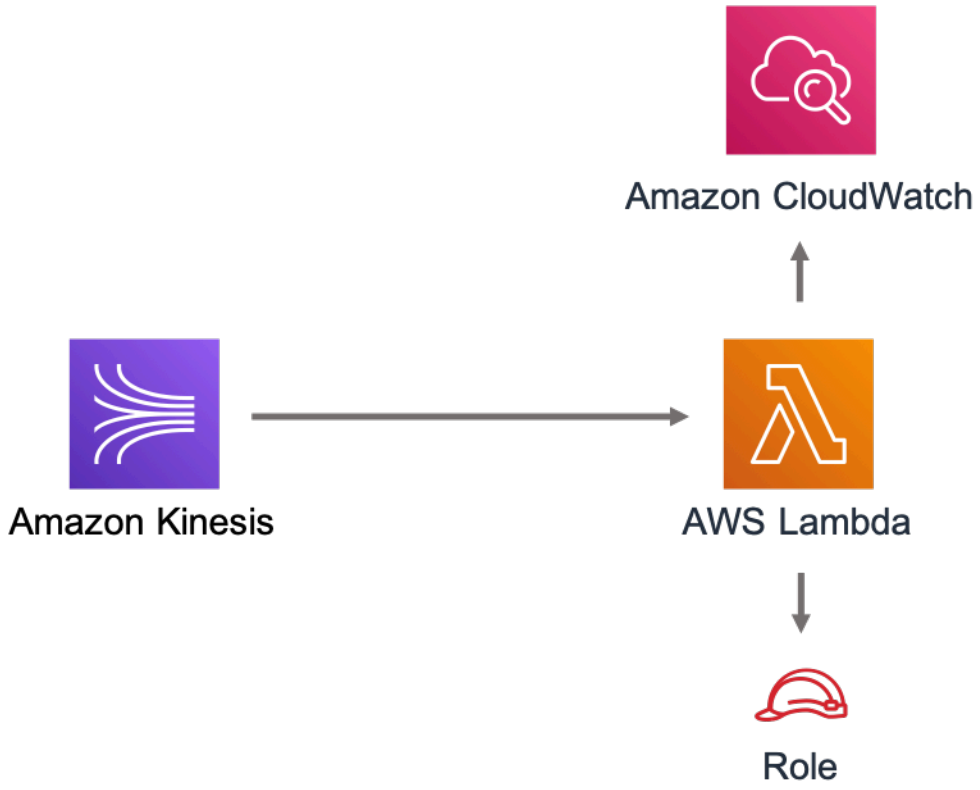
Amazon Kinesis Stream

- Configure el rol de IAM de acceso mínimo con privilegios para Kinesis Stream.
- Habilite el cifrado del lado del servidor para Kinesis Stream mediante la clave KMS administrada por AWS.
- Implemente las alarmas de CloudWatch de las mejores prácticas para Kinesis Stream.

Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Active el rastreo de X-Ray.
- Habilitar funciones de control de errores: habilite bisect on function Error; establezca la antigüedad máxima de registro predeterminada (24 horas); establezca el intento máximo de reintento predeterminado (500); e implemente la cola de letras muertas SQS como destino en caso de fallo.
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-kinesisstreams-lambda

aws-lambda-dynamodb

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_lambda_dynamodb`
Mecanografía	`@aws-solutions-constructs/aws-lambda-dynamodb`
Java	`software.amazon.awsconstructs.services.lambdadynamodb`

Overview

Este constructo de soluciones de AWS implementa la función AWS Lambda y la tabla de Amazon DynamoDB con permisos de mínimo privilegio.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { LambdaToDynamoDBProps,  LambdaToDynamoDB } from '@aws-solutions-constructs/
aws-lambda-dynamodb';

const props: LambdaToDynamoDBProps = {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    }
};

new LambdaToDynamoDB(this, 'test-lambda-dynamodb-stack', props);
```

Initializer

```
new LambdaToDynamoDB(scope: Construct, id: string, props: LambdaToDynamoDBProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`LambdaToDynamoDBProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si `unexistingLambdaObj` se encuentra a su disposición.
¿DynamotableProps?	`dynamodb.TableProps`	Los accesorios proporcionados por el usuario opcionales para anular los accesorios predeterminados para la tabla de DynamoDB

Nombre	Tipo	Descripción
ExistingTableObj?	`dynamodb.Table`	Instancia existente del objeto de tabla DynamoDB, proporcionando tanto esto como`dynamoTableProps` provocará un error.
¿Permisos de tabla?	`string`	Permisos de tabla opcionales que se concederán a la función Lambda. Puede que se especifique una de las siguientes opciones: `All`,`Read`,`ReadWrite` , o bien`Write`.
TableEnvironmentVariableName?	`string`	Nombre opcional para el conjunto de variables de entorno de tabla DynamoDB para la función Lambda.

Nombre	Tipo	Descripción
¿Existe VPC?	`ec2.IVpc`	Una VPC opcional existente en la que se debe implement ar este patrón. Cuando se implementa en una VPC, la función Lambda utilizará ENI en la VPC para acceder a los recursos de red y se creará un extremo de puerta de enlace en la VPC para Amazon DynamoDB. Si se proporciona una VPC existente, el`deployVpc` no se puede`true`. Utiliza`ec2.IVpc`para permitir que los clientes proporcionen VPC que existen fuera de la pila mediante el comando`ec2.Vpc.fromLookup()` Método de.

Nombre	Tipo	Descripción
¿VPCProps?	`ec2.VpcProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas de la nueva VPC.`enableDns Hostnames` ,`enableDns Support` ,`natGateways` , `ysubnetConfiguratio n` están establecidos por el patrón, por lo que cualquier valor para esas propiedad es suministradas aquí será anulado. Si`deployVpc` no es`true`entonces esta propiedad será ignorada.

Nombre	Tipo	Descripción
¿Implementar VPC?	`boolean`	Si se crea una nueva VPC basada en`vpcProps`en el que desplegar este patrón. Si se establece esto en true, se implementará la VPC mínima y más privada para ejecutar el patrón: • Una subred aislada en cada zona de disponibilidad utilizada por el programa CDK • `enableDnsHostnames` y`enableDns Support` ambos se establecerán en`true` Si esta propiedad es`true`, luego`existingVpc` No se puede especificar . El valor predeterminado es `false`.

Propiedades de patrón

Nombre	Tipo	Descripción
DynamoTable	`dynamodb.Table`	Devuelve una instancia de la tabla DynamoDB creada por el patrón.
Lambdafunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.

Nombre	Tipo	Descripción
¿VPC?	`ec2.IVpc`	Devuelve una interfaz en la VPC utilizada por el patrón (si la hay). Esta puede ser una VPC creada por el patrón o la VPC suministrada al construct or del patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

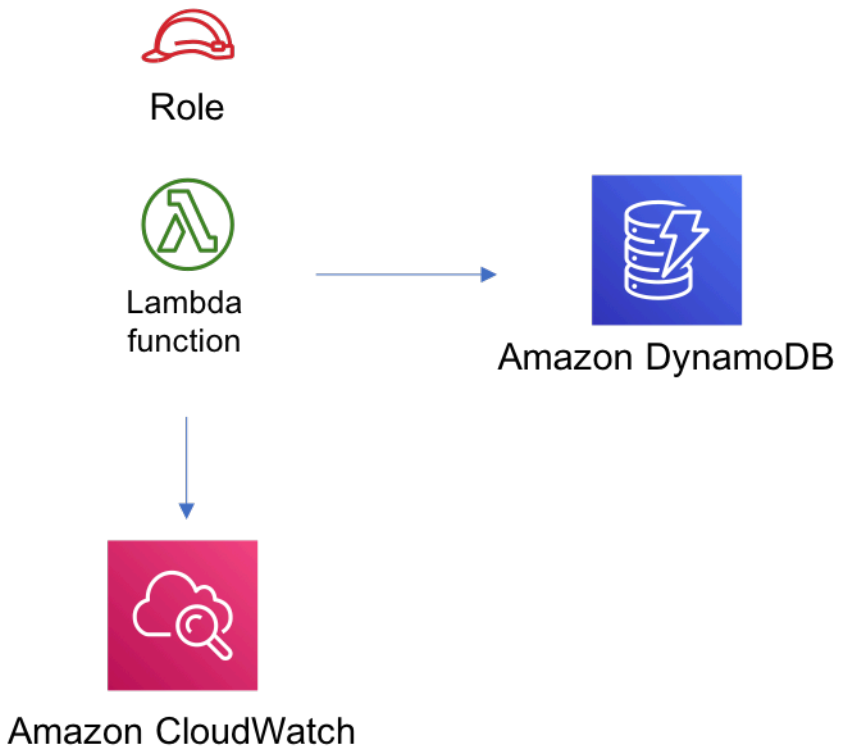
Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Active el rastreo de X-Ray.
- Configuración de las variables de entorno:
  - `DDB_TABLE_NAME` (predeterminado)
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Amazon DynamoDB Tabla

- Establezca el modo de facturación para la tabla de DynamoDB en Bajo demanda (pago por solicitud).
- Habilite el cifrado del lado del servidor para DynamoDB Table mediante la clave KMS administrada por AWS.
- Crea una clave de partición llamada 'id' para DynamoDB Table.
- Conservar la tabla al eliminar la pila CloudFormation.
- Habilite las copias de seguridad continuas y la recuperación a un momento dado.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-lambda-dynamodb

aws-lambda-elasticsearch-kibana

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_lambda_elasticsearch_kibana`
Mecanografiado	`@aws-solutions-constructs/aws-lambda-elasticsearch-kibana`
Java	`software.amazon.awsconstructs.services.lambdaelasticsearchkibana`

Overview

Este constructo de soluciones de AWS implementa una función de AWS Lambda y un dominio de Amazon Elasticsearch Service con permisos con menos privilegios.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { LambdaToElasticSearchAndKibana } from '@aws-solutions-constructs/aws-lambda-elasticsearch-kibana';
import { Aws } from "@aws-cdk/core";

const lambdaProps: lambda.FunctionProps = {
    runtime: lambda.Runtime.NODEJS_14_X,
    // This assumes a handler function in lib/lambda/index.js
    code: lambda.Code.fromAsset(`${__dirname}/lambda`),
    handler: 'index.handler'
};

new LambdaToElasticSearchAndKibana(this, 'test-lambda-elasticsearch-kibana', {
    lambdaFunctionProps: lambdaProps,
    domainName: 'test-domain',
    // TODO: Ensure the Cognito domain name is globally unique
    cognitoDomainName: 'globallyuniquedomain' + Aws.ACCOUNT_ID;
});
```

Initializer

```
new LambdaToElasticSearchAndKibana(scope: Construct, id: string, props:
 LambdaToElasticSearchAndKibanaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`LambdaToElasticSearchAndKibanaProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si un`existingLambdaObj` se encuentra.
¿EsDomainProps?	`elasticsearch.CfnDomainProps`	Los accesorios proporcionados por el usuario opcionales para anular los accesorios predeterminados para Amazon Elasticsearch Service

Nombre	Tipo	Descripción
domainName	`string`	Nombre de dominio para Cognito y Amazon Elasticsearch Service
CognitoDomainName?	`string`	Nombre de dominio opcional de Cognito. Si se proporciona, se utilizará para el dominio Cognito, y`domainName` se utilizará para el dominio Elasticsearch.
CreateCloudWatchalarms	`boolean`	Si desea crear alarmas recomendadas de CloudWatch.
DomainEndPointEnvironmentVariableName?	`string`	Nombre opcional para el conjunto de variables de entorno de extremo de dominio ElasticSearch para la función Lambda.

Propiedades de patrón

Nombre	Tipo	Descripción
¿Cloudwatchalarms?	`cloudwatch.Alarm[]`	Devuelve una lista de una o más alarmas de CloudWatch creadas por el patrón.
ElasticSearchDomain	`elasticsearch.CfnDomain`	Devuelve una instancia del dominio Elasticsearch creado por el patrón.

Nombre	Tipo	Descripción
ElasticSearchDomainRole	`iam.Role`	Devuelve una instancia del rol de IAM creado por el patrón para el dominio Elasticsearch.
IdentityPool	`cognito.CfnIdentityPool`	Devuelve una instancia del grupo de identidades de Cognito creado por el patrón.
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.
userPool	`cognito.UserPool`	Devuelve una instancia del grupo de usuarios de Cognito creado por el patrón.
UserPoolClient	`cognito.UserPoolClient`	Devuelve una instancia del cliente de grupo de usuarios de Cognito creado por el patrón.

Lambda function

Este patrón requiere una función Lambda que pueda publicar datos en el servicio Elasticsearch desde la secuencia de DynamoDB. Se proporciona una función de muestraAquí.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Active el rastreo de X-Ray.

- Configuración de las variables de entorno:
  - `DOMAIN_ENDPOINT` (predeterminado)
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

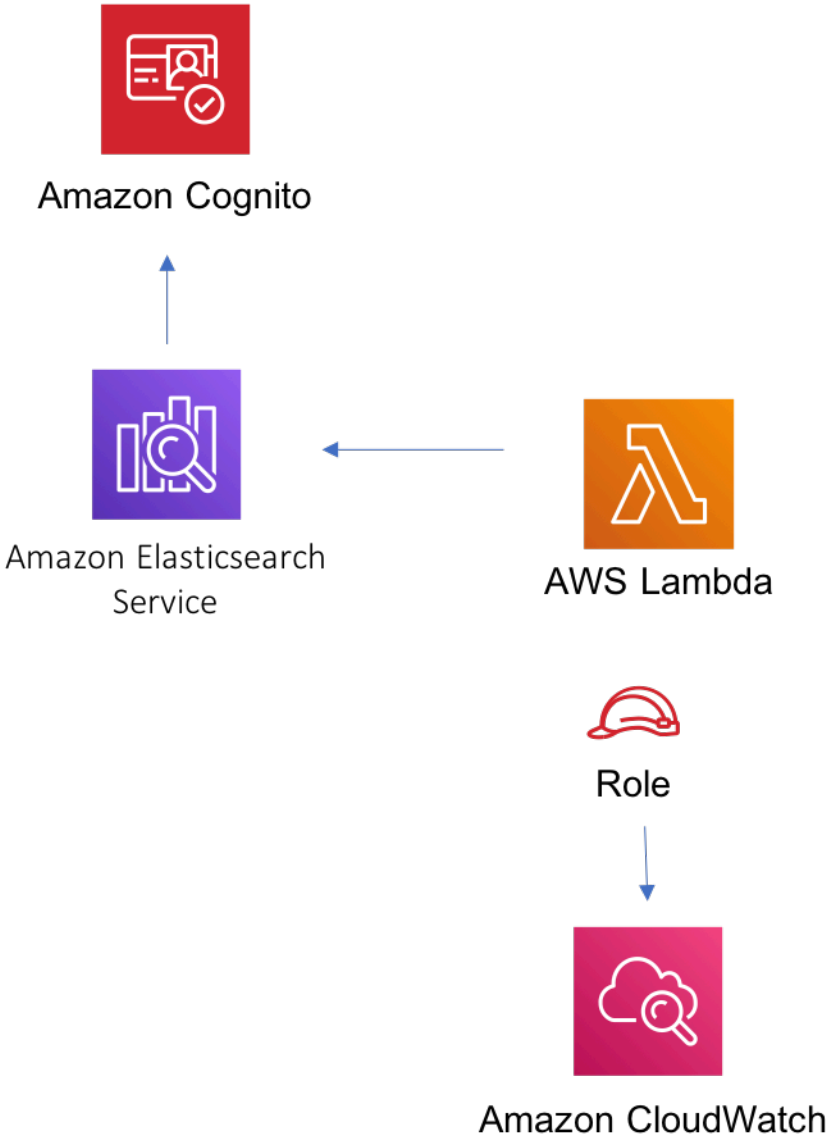
Amazon Cognito

- Establezca la directiva de contraseñas para grupos de usuarios.
- Aplique el modo de seguridad avanzada para los grupos de usuarios.

Amazon Elasticsearch Service

- Implemente las prácticas recomendadas de CloudWatch Alarms para el dominio Elasticsearch.
- Proteja el acceso al panel de Kibana con Grupos de usuarios de Cognito.
- Habilite el cifrado del lado del servidor para el dominio Elasticsearch mediante la clave KMS administrada por AWS.
- Habilite el cifrado de nodo a nodo para el dominio Elasticsearch.
- Configure el clúster para el dominio Amazon ES.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-lambda-elasticsearch-kibana

aws-lambda-s3

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_lambda_s3`
Mecanografía	`@aws-solutions-constructs/aws-lambda-s3`
Java	`software.amazon.awsconstructs.services.lambdas3`

Overview

Este constructo de soluciones de AWS implementa una función de AWS Lambda conectada a un bucket de Amazon S3.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { LambdaToS3 } from '@aws-solutions-constructs/aws-lambda-s3';

new LambdaToS3(this, 'LambdaToS3Pattern', {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
```

```
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    }
});
```

Initializer

```
new LambdaToS3(scope: Construct, id: string, props: LambdaToS3Props);
```

Parámetros

- scope`Construct`
- id`string`
- props`LambdaToS3Props`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si un`existingLambdaObj` se encuentra a su disposición.

Nombre	Tipo	Descripción
¿ExistenteBuckeTobj?	`s3.IBucket`	Instancia existente del objeto S3 Bucket. Si se proporcio na esto, entonces también proporcionabucketProps es un error.
¿BucketProps?	`s3.BucketProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas del depósito. Se pasa por alto si unexistingBucketObj se encuentra a su disposición.
BucketPermisos?	`string[]`	Permisos de depósito opcionales para conceder a la función Lambda. Pueden especificarse una o varias de las siguiente s:Delete,Put,Read,ReadWrite ,Write.

Nombre	Tipo	Descripción
¿Existe VPC?	`ec2.IVpc`	Una VPC opcional existente en la que se debe implement ar este patrón. Cuando se implementa en una VPC, la función Lambda utilizará ENI en la VPC para acceder a los recursos de red y se creará un extremo de interfaz en la VPC para Amazon SQS. Si se proporciona una VPC existente, el`deployVpc` no se puede`true`. Esto utiliza`ec2.IVpc`para permitir que los clientes suministren VPC que existen fuera de la pila mediante el comando`ec2.Vpc.fromLookup()` Método de.

Nombre	Tipo	Descripción
¿Implementar VPC?	`boolean`	Si se crea una nueva VPC basada en`vpcProps`en el que desplegar este patrón. Configuración de esto en`true`implementará la VPC mínima y más privada para ejecutar el patrón: • Una subred aislada en cada zona de disponibilidad utilizada por el programa CDK. • `enableDnsHostnames` y`enableDnsSupport` ambos se establecerán en`true`. Si esta propiedad es`true`, luego`existingVpc` No se puede especificar . El valor predeterminado es `false`.

Nombre	Tipo	Descripción
¿VPCProps?	`ec2.VpcProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas de la nueva VPC.`enableDns Hostnames` ,`enableDns Support` ,`natGatewa ys` `ysubnetConfiguratio n` están establecidos por el patrón, por lo que cualquier valor para esas propiedad es suministradas aquí será anulado. Si`deployVpc` no es`true`Esta propiedad se pasará por alto.
BucketEnvironmentVariableNa me?	`string`	Nombre opcional para el conjunto de variables de entorno de cucharón S3 para la función Lambda.

Propiedades de patrón

Nombre	Tipo	Descripción
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.
¿S3 Bucket?	`s3.Bucket`	Devuelve una instancia del bucket S3 creado por el patrón.

Nombre	Tipo	Descripción
¿S3LoggingBucket?	`s3.Bucket`	Devuelve una instancia del depósito de registro creado por el patrón para el bucket S3.
¿VPC?	`ec2.IVpc`	Devuelve una instancia de la VPC utilizada por el patrón (si la hay). Esta puede ser una VPC creada por el patrón o la VPC suministrada al construct or del patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Habilitar el X-Ray streo
- Configuración de las variables de entorno:
  - `S3_BUCKET_NAME` (predeterminado)
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Bucket de Amazon S3

- Configurar el registro de acceso para S3 Bucket.
- Habilite el cifrado del lado del servidor para S3 Bucket mediante la clave KMS administrada por AWS.
- Active el control de versiones para S3 Bucket.
- No permita el acceso público para S3 Bucket.
- Mantenga el Bucket S3 cuando elimine la pila CloudFormation.

- Aplicación del cifrado de los datos en tránsito.
- Aplica una regla de ciclo de vida para mover versiones de objetos no actuales al almacenamiento de Glacier después de 90 días.

Architecture

GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-lambda-s3

aws-lambda-ssmstringparameter

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_lambda_ssm_string_parameter`
Mecanografía	`@aws-solutions-constructs/aws-lambda-ssmstringparameter`
Java	`software.amazon.awsconstructs.services.lambdassmstringparameter`

Overview

Este constructo de soluciones de AWS implementa la función AWS Lambda y el parámetro AWS Systems Manager Parameter Store String con los permisos menos privilegiados.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
const { LambdaToSsmstringparameterProps,  LambdaToSsmstringparameter } from '@aws-solutions-constructs/aws-lambda-ssmstringparameter';

const props: LambdaToSsmstringparameterProps = {
    lambdaFunctionProps: {
      runtime: lambda.Runtime.NODEJS_14_X,
      // This assumes a handler function in lib/lambda/index.js
      code: lambda.Code.fromAsset(`${__dirname}/lambda`),
      handler: 'index.handler'
    },
    stringParameterProps: { stringValue: "test-string-value" }
};

new LambdaToSsmstringparameter(this, 'test-lambda-ssmstringparameter-stack', props);
```

Initializer

```
new LambdaToSsmstringparameter(scope: Construct, id: string, props:
 LambdaToSsmstringparameterProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`LambdaToSsmstringparameterProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si see`existingLambdaObj` se encuentra.
ExistingStringParameterObj?	`ssm.StringParameter`	Instancia existente del objeto de parámetro SSM String, proporcionando tanto esto como`stringPar`

Nombre	Tipo	Descripción
		`ameterProps` provocará un error.
StringParameterProps?	`ssm.StringParameterProps`	El usuario ha proporcionado accesorios opcionales para anular los apoyos predeterminados para el parámetro SSM String. Si`existingStringParameterObj` no está establecido,`stringParameterProps` es obligatorio. El único soporte`ssm.StringParameterProps.type` es`STRING`si se proporciona un valor diferente, se anulará.
StringParameterEnvironmentVariableName?	`string`	Nombre opcional para el conjunto de variables de entorno del parámetro SSM String para la función Lambda.

Nombre	Tipo	Descripción
¿Existe VPC?	`ec2.IVpc`	Una VPC opcional existente en la que se debe implement ar este patrón. Cuando se implementa en una VPC, la función Lambda utilizará ENI en la VPC para acceder a los recursos de red y se creará un extremo de interfaz en el parámetro VPC for AWS Systems Manager. Si se proporciona una VPC existente, el`deployVpc` no se puede`true`. Usa`ec2.IVpc`para permitir que los clientes suministr en VPC que existen fuera de la pila mediante el comando`ec2.Vpc.fromLookup()` Método de.

Nombre	Tipo	Descripción
¿VPCProps?	`ec2.VpcProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas de la nueva VPC.`enableDns Hostnames` ,`enableDns Support` ,`natGatewa ys` `ysubnetConfiguratio n` están establecidos por el patrón, por lo que cualquier valor para esas propiedad es suministradas aquí será anulado. Si`deployVpc` no es`true`entonces esta propiedad será ignorada.

Nombre	Tipo	Descripción
¿Implementar VPC?	`boolean`	Si se crea una VPC nueva basada en`vpcProps`en el que desplegar este patrón. Configuración de esto en`true`implementará la VPC mínima y más privada para ejecutar el patrón: • Una subred aislada en cada zona de disponibilidad utilizada por el programa CDK. • `enableDnsHostnames` y`enableDnsSupport` ambos se establecerán en`true`. Si esta propiedad está establecida en`true`, luego`existingVpc` No se puede especificar . El valor predeterminado es `false`.
StringParameterPermissions?	`string`	Permisos opcionales del parámetro SSM String para conceder a la función Lambda. Puede especificarse una de las siguientes opciones: `Read`,`ReadWrite` .

Propiedades de patrón

Nombre	Tipo	Descripción
Lambdafunction	`lambda.Function`	Devuelve una instancia de `lambda.Function` creado por el componente fijo.
StringParameter	`ssm.StringParameter`	Devuelve una instancia de `ssm.StringParameter` creado por el componente fijo.
¿VPC?	`ec2.IVpc`	Devuelve una interfaz en la VPC utilizada por el patrón (si la hay). Esta puede ser una VPC creada por el patrón o la VPC suministrada al constructor del patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

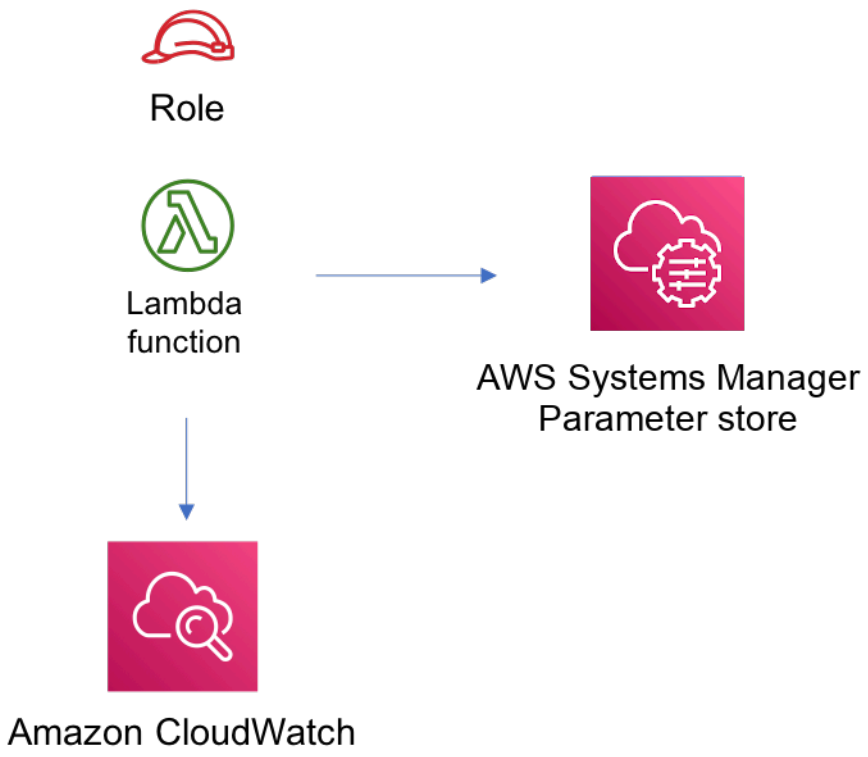
Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Active el rastreo de X-Ray.
- Configuración de las variables de entorno:
  - `SSM_STRING_PARAMETER_NAME` (predeterminado)
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

String de almacén de parámetros de Amazon Systems Manager

- Habilite el acceso de sólo lectura para la función AWS Lambda asociada.
- Crea un nuevo parámetro de cadena SSM con los valores proporcionados.
- Conservar el parámetro SSM String al eliminar la pila CloudFormation.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-lambda-ssmstringparameter

aws-lambda-sagemakerendpoint

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_lambda_sagemakerendpoint`
Mecanografía	`@aws-solutions-constructs/aws-lambda-sagemakerendpoint`
Java	`software.amazon.awsconstructs.services.lambdasagemakerendpoint`

Overview

Este constructo de soluciones de AWS implementa una función de AWS Lambda conectada a un punto final de Amazon Sagemaker.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { Duration } from '@aws-cdk/core';
import * as lambda from '@aws-cdk/aws-lambda';
import {
  LambdaToSagemakerEndpoint,
  LambdaToSagemakerEndpointProps,
} from '@aws-solutions-constructs/aws-lambda-sagemakerendpoint';

const constructProps: LambdaToSagemakerEndpointProps = {
  modelProps: {
    primaryContainer: {
```

```
      image: '{{AccountId}}.dkr.ecr.{{region}}.amazonaws.com/linear-learner:latest',
      modelDataUrl: 's3://{{bucket-name}}/{{prefix}}/model.tar.gz',
    },
  },
  lambdaFunctionProps: {
    runtime: lambda.Runtime.PYTHON_3_8,
    // This assumes a handler function in lib/lambda/index.py
    code: lambda.Code.fromAsset(`${__dirname}/lambda`),
    handler: 'index.handler',
    timeout: Duration.minutes(5),
    memorySize: 128,
  },
};

new LambdaToSagemakerEndpoint(this, 'LambdaToSagemakerEndpointPattern',
 constructProps);
```

Initializer

```
new LambdaToSagemakerEndpoint(scope: Construct, id: string, props:
 LambdaToSagemakerEndpointProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`LambdaToSagemakerEndpointProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFun`

Nombre	Tipo	Descripción
		`ctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda.
ExistenteSageMakerendPointObj?	`sagemaker.CfnEndpoint`	Un Enpoint Sagemaker existente opcional que se va a utilizar. Proporcionando tanto esto como`endpointProps` provocará un error.
¿ModelProps?	`sagemaker.CfnModelProps` | `any`	Propiedades proporcionadas por el usuario para anular las propiedades predeterminadas para el modelo de Sagemaker. Como mínimo`modelProps.primaryContainer` para crear un modelo. De forma predeterminada, el patrón creará un rol con los permisos mínimos requeridos, pero el cliente puede proporcionar un rol personalizado con capacidades adicionales usando`modelProps.executionRoleArn`.

Nombre	Tipo	Descripción
EndPointConfigProps?	`sagemaker.CfnEndpointConfigProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la configuración de Sagemaker Endpoint.
¿EndpointProps?	`sagemaker.CfnEndpointProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas para el extremo de Sagemaker.
¿Existe VPC?	`ec2.IVpc`	VPC opcional existente en la que se debe implementar esta construcción. Cuando se implementan en una VPC, la función Lambda y Sagemaker Endpoint usarán ENI en la VPC para acceder a los recursos de red. Se creará un extremo de interfaz en la VPC para Amazon Sagemaker Runtime y Amazon S3 VPC Endpoint. Si se proporciona una VPC existente, el `deployVpc` propiedad no puede ser `true`.

Nombre	Tipo	Descripción
¿VPCProps?	`ec2.VpcProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas de la nueva VPC.`enableDns Hostnames` ,`enableDns Support` ,`natGatewa ys` `ysubnetConfiguratio n` están establecidos por el constructo, por lo que cualquier valor para las propiedades suministr adas aquí será anulado. `SideployVpc` no es`true`, esta propiedad será ignorada.

Nombre	Tipo	Descripción
¿Implementar VPC?	`boolean`	Si se crea una VPC nueva basada en`vpcProps`en el que desplegar este patrón. Configuración de esto en`true`implementará la VPC mínima y más privada para ejecutar el patrón: • Una subred aislada en cada zona de disponibilidad utilizada por el programa CDK. • `enableDnsHostnames` y`enableDnsSupport` ambos se establecerán en`true`. Si esta propiedad está establecida en`true`, luego`existingVpc` No se puede especificar . El valor predeterminado es `false`.
SageMakerEnvironmentVariableName?	`string`	Nombre opcional para el conjunto de variables de entorno de punto final de SageMaker para la función Lambda.

Propiedades de patrón

Nombre	Tipo	Descripción
Lambdafunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.
SageMakerEndPoint	`sagemaker.CfnEndpoint`	Devuelve una instancia del extremo de Sagemaker creado por el patrón.
SageMakerEndPointConfig?	`sagemaker.CfnEndpointConfig`	Devuelve una instancia del SageMaker EndpointConfig creado por el patrón, si `existingSagemakerEndpointObj` no se proporciona.
¿Modelo SageMakerModel?	`sagemaker.CfnModel`	Devuelve una instancia del modelo de Sagemaker creado por el patrón, si `existingSagemakerEndpointObj` no se proporciona.
¿VPC?	`ec2.IVpc`	Devuelve una instancia de la VPC creada por el patrón, si `deployVpc` es `true`, o si `existingVpc` toma el valor.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

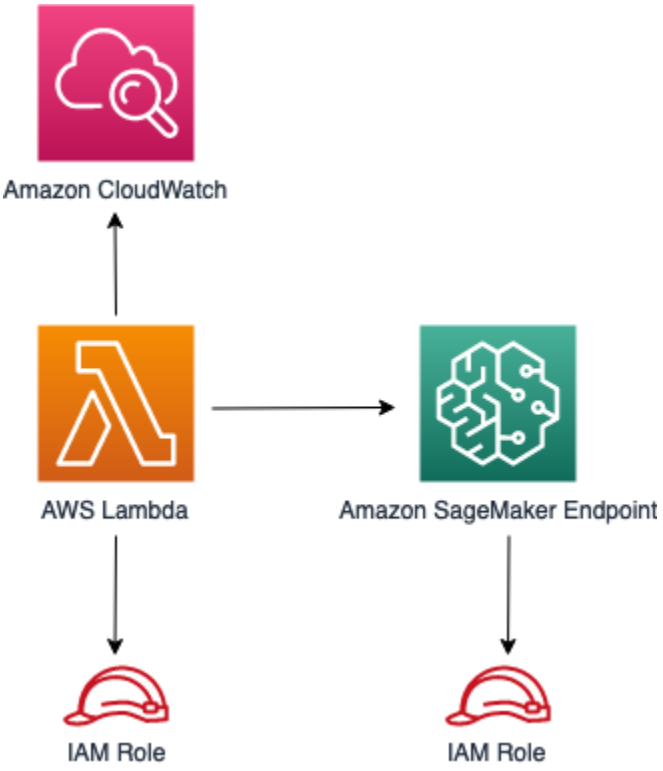
Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Permita que la función invoque el extremo de Sagemaker para inferencias.
- Configure la función para acceder a los recursos de la VPC, donde se implementa el extremo de Sagemaker.
- Active Rastreo de X-Ray.
- Configuración de las variables de entorno:
  - `SAGEMAKER_ENDPOINT_NAME` (predeterminado)
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Punto de enlace de Amazon Sagemaker

- Configure privilegios limitados para crear recursos de Sagemaker.
- Implemente el modelo de Sagemaker, EndPointConfig y endpoint.
- Configure el extremo de Sagemaker que se va a implementar en una VPC.
- Implemente S3 VPC Endpoint y Sagemaker Runtime VPC Interface.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-lambda-sagemakerendpoint

aws-lambda-secretsmanager

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_lambda_secretsmanager`
Mecanografía	`@aws-solutions-constructs/aws-lambda-secretsmanager`
Java	`software.amazon.awsconstructs.services.lambdasecretsmanager`

Overview

Este constructo de soluciones de AWS implementa la función AWS Lambda y el secreto de AWS Secrets Manager con los permisos menos privilegiados.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
const { LambdaToSecretsmanagerProps,  LambdaToSecretsmanager } from '@aws-solutions-constructs/aws-lambda-secretsmanager';

const props: LambdaToSecretsmanagerProps = {
    lambdaFunctionProps: {
      runtime: lambda.Runtime.NODEJS_14_X,
      // This assumes a handler function in lib/lambda/index.js
      code: lambda.Code.fromAsset(`${__dirname}/lambda`),
      handler: 'index.handler'
    },
};

new LambdaToSecretsmanager(this, 'test-lambda-secretsmanager-stack', props);
```

Initializer

```
new LambdaToSecretsmanager(scope: Construct, id: string, props:
 LambdaToSecretsmanagerProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`LambdaToSecretsmanagerProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	El usuario proporcionó accesorios para anular los accesorios predeterminados para la función Lambda.
¿SecretProps?	`secretsmanager.SecretProps`	El usuario opcional proporcionó accesorios para anular los accesorios predeterminados para Secrets Manager.
¿ExistenteSecretObj?	`secretsmanager.Secret`	Instancia existente de Secrets Manager objeto secreto, Si esto se establece entonces el`secretProps` se ignora.
¿GrantWriteAccess?	`boolean`	Acceso de escritura opcional al secreto para la función

Nombre	Tipo	Descripción
		Lambda (sólo lectura por defecto).
SecretenEnvironmentVariable Name?	`string`	Nombre opcional para el conjunto de variables de entorno secretas de Secrets Manager para la función Lambda.
¿Existe VPC?	`ec2.IVpc`	Una VPC opcional existente en la que se debe implement ar este patrón. Cuando se implementa en una VPC, la función Lambda utilizará ENI en la VPC para acceder a los recursos de red y se creará un extremo de interfaz en la VPC para AWS Secrets Manager. Si se proporciona una VPC existente, el`deployVpc` no se puede`true`. Esto usa`ec2.IVpc`para permitir que los clientes suministr en VPC que existen fuera de la pila mediante el comando`ec2.Vpc.fromLookup()` Método de.

Nombre	Tipo	Descripción
¿VPCProps?	`ec2.VpcProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas de la nueva VPC.`enableDns Hostnames` ,`enableDns Support` ,`natGateways` , `ysubnetConfiguratio n` están establecidos por el patrón, por lo que cualquier valor para esas propiedad es suministradas aquí será anulado. Si`deployVpc` no es`true`esta propiedad se ignorará.

Nombre	Tipo	Descripción
¿Implementar VPC?	`boolean`	Si se crea una VPC nueva basada en`vpcProps`en el que desplegar este patrón. Configuración de esto en`true`implementará la VPC mínima y más privada para ejecutar el patrón: • Una subred aislada en cada zona de disponibilidad utilizada por el programa CDK • `enableDnsHostnames` y`enableDnsSupport` ambos se establecerán en`true` Si esta propiedad es`true`, luego`existingVpc` No se puede especificar . El valor predeterminado es `false`.

Propiedades de patrón

Nombre	Tipo	Descripción
Lambdafunction	`lambda.Function`	Devuelve una instancia de`lambda.Function` creado por el componente fijo.
secreta	`secretsmanager.Secret`	Devuelve una instancia de`secretsmanager.Sec`

Nombre	Tipo	Descripción
		`ret` creado por el component e fijo.
¿VPC?	`ec2.IVpc`	Devuelve una interfaz en la VPC utilizada por el patrón (si la hay). Esta puede ser una VPC creada por el patrón o la VPC suministrada al construct or del patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

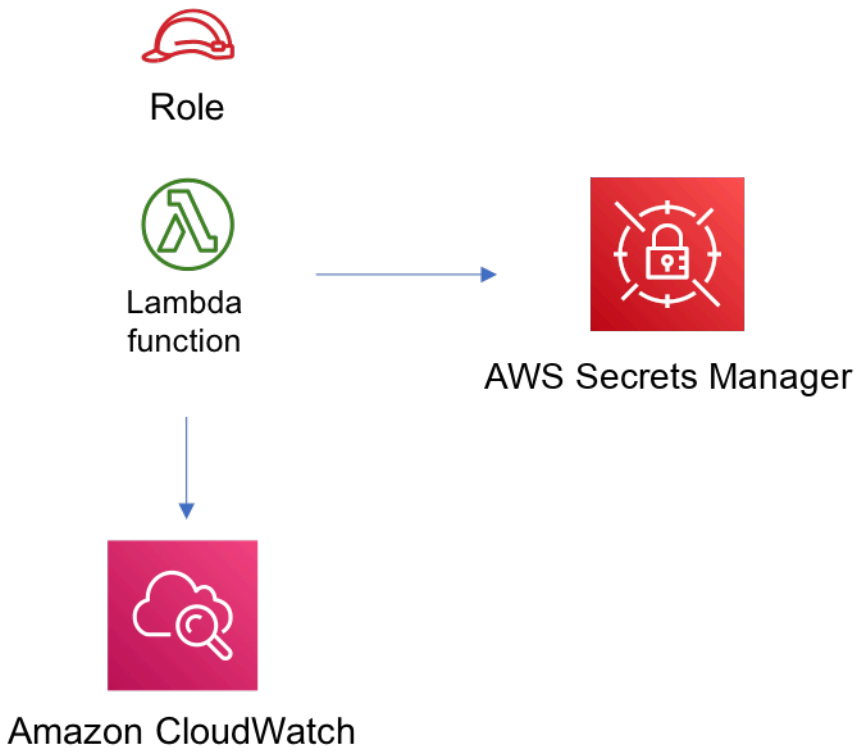
Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Active el rastreo de X-Ray.
- Configuración de las variables de entorno:
  - (predeterminado) SECRET_ARN que contiene el ARN del secreto como devuelto por CDKSecretArnPropiedad
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Secreto de Amazon Secrets Manager

- Habilitar el acceso de sólo lectura para la función AWS Lambda asociada
- Habilitar el cifrado del lado del servidor mediante una clave KMS predeterminada para la cuenta y la región
- Crea un nuevo secerto:
  - (predeterminado) nombre aleatorio
  - (predeterminado) valor aleatorio
- Conservar el secreto al eliminar la pila de CloudFormation

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-lambda-secretsmanager

aws-lambda-sns

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_lambda_sns`
Mecanografía	`@aws-solutions-constructs/aws-lambda-sns`
Java	`software.amazon.awsconstructs.services.lambdasns`

Overview

Este constructo de soluciones de AWS implementa una función de AWS Lambda conectada a un tema de Amazon SNS.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { LambdaToSns, LambdaToSnsProps } from "@aws-solutions-constructs/aws-lambda-
sns";

new LambdaToSns(this, 'test-lambda-sns', {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    }
});
```

Initializer

```
new LambdaToSns(scope: Construct, id: string, props: LambdaToSnsProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`LambdaToSnsProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Si se pasa por alto un`existingLambdaObj` se encuentra.
¿ExistentetoPicobj?	`sns.Topic`	Instancia existente del objeto SNS Topic, proporcionando tanto esto como`topicProps` provocará un error.
¿TopicProps?	`sns.TopicProps`	Propiedades proporcionadas por el usuario opcionales para anular las propiedades predeterminadas para el tema SNS.

Nombre	Tipo	Descripción
¿Existe VPC?	`ec2.IVpc`	Una VPC opcional existente en la que se debe implementar este patrón. Cuando se implementa en una VPC, la función Lambda utilizará ENI en la VPC para acceder a los recursos de red y se creará un extremo de interfaz en la VPC para Amazon SQS. Si se proporciona una VPC existente, el`deployVpc` no se puede`true`. Para ello se utiliza`ec2.IVpc`para permitir que los clientes suministren VPC que existen fuera de la pila mediante el comando`ec2.Vpc.fromLookup()` Método de.

Nombre	Tipo	Descripción
¿Implementar VPC?	`boolean`	Si se va a crear una VPC nueva basada en`vpcProps`en el que desplegar este patrón. Si se establece este valor en`true`implementará la VPC mínima y más privada para ejecutar el patrón: • Una subred aislada en cada zona de disponibilidad utilizada por el programa CDK. • `enableDnsHostnames` y`enableDnsSupport` ambos se establecerán en`true`. Si esta propiedad es`true`, luego`existingVpc` No se puede especificar . El valor predeterminado es `false`.

Nombre	Tipo	Descripción
¿VPCProps?	`ec2.VpcProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas de la nueva VPC.`enableDns Hostnames` ,`enableDns Support` ,`natGatewa ys` `ysubnetConfiguratio n` están establecidos por el patrón, por lo que cualquier valor para esas propiedad es suministradas aquí será anulado. Si`deployVpc` no es`true`Si se pasa por alto esta propiedad.
TopArnEnvironmentV ariableName?	`string`	Nombre opcional para el conjunto de variables de entorno ARN del tema SNS para la función Lambda.
TopNameEnvironment VariableName?	`string`	Nombre opcional para el conjunto de variables de entorno de nombre de tema SNS para la función Lambda.

Propiedades de patrón

Nombre	Tipo	Descripción
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.

Nombre	Tipo	Descripción
SNSTopic	`sns.Topic`	Devuelve una instancia del tema SNS creado por el patrón.
¿VPC?	`ec2.IVpc`	Devuelve una instancia de la VPC utilizada por el patrón (si la hay). Esta puede ser una VPC creada por el patrón o la VPC suministrada al construct or del patrón.

Configuración predeterminada

La implementación inmediata del componente fijo sin anulación establecerá los siguientes valores predeterminados:

Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Active el rastreo de X-Ray.
- Configuración de las variables de entorno:
  - `SNS_TOPIC_NAME` (predeterminado)
  - `SNS_TOPIC_ARN` (predeterminado)
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Tema de Amazon SNS

- Configurar permisos de acceso de privilegios mínimos para el tema SNS.
- Habilite el cifrado del lado del servidor mediante la clave KMS administrada por AWS.
- Aplicación del cifrado de los datos en tránsito.

Architecture

GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-lambda-sns

aws-lambda-sqs

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersionado semánticoModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_lambda_sqs`

Idioma	Paquete
Mecanografía	`@aws-solutions-constructs/aws-lambda-sqs`
Java	`software.amazon.awsconstructs.services.lambdasqs`

Overview

Este constructo de soluciones de AWS implementa una función de AWS Lambda conectada a una cola de Amazon SQS.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { LambdaToSqs, LambdaToSqsProps } from "@aws-solutions-constructs/aws-lambda-sqs";

new LambdaToSqs(this, 'LambdaToSqsPattern', {
   lambdaFunctionProps: {
       runtime: lambda.Runtime.NODEJS_14_X,
       // This assumes a handler function in lib/lambda/index.js
       code: lambda.Code.fromAsset(`${__dirname}/lambda`),
       handler: 'index.handler'
   }
});
```

Initializer

```
new LambdaToSqs(scope: Construct, id: string, props: LambdaToSqsProps);
```

Parámetros

- scope`Construct`

- `idstring`
- propsLambdaToSqsProps

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	lambda.Function	Una función Lambda opcional existente que se utilizará en lugar de la función predeterminada. Proporcionando tanto esto como `lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	lambda.FunctionProps	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda.
ExistingQueueObj?	sqs.Queue	Una cola SQS opcional existente que se utilizará en lugar de la cola predeterminada. Proporcionando tanto esto como `queueProps` provocará un error.
¿Props en cola?	sqs.QueueProps	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la cola SQS.
¿EnableQueuePurging?	`boolean`	Si se deben conceder permisos adicionales a la función Lambda, lo que le

Nombre	Tipo	Descripción
		permite depurar la cola SQS. El valor predeterminado es `false`.
DeployDeadLetterQueue?	`boolean`	Si se crea una cola secundari a para utilizarse como cola de mensajes fallidos. El valor predeterminado es `true`.
DeadletterQueueProps?	`sqs.QueueProps`	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para la cola de letras muertas. Solo se usa si el`deployDeadLetterQu eue` toma el valor verdadero.
¿maxReceiveCount?	`number`	El número de veces que un mensaje se puede quitar sin éxito antes de que se traslade a la cola de mensajes fallidos. El valor predeterminado es `15`.

Nombre	Tipo	Descripción
¿Existe VPC?	`ec2.IVpc`	Una VPC opcional existente en la que se debe implement ar este patrón. Cuando se implementa en una VPC, la función Lambda utilizará ENI en la VPC para acceder a los recursos de red y se creará un extremo de interfaz en la VPC para Amazon SQS. Si se proporciona una VPC existente, el`deployVpc` no se puede`true`. Una`ec2.IVpc`se utiliza para permitir que los clientes proporcionen VPC que existen fuera de la pila mediante el comando`ec2.Vpc.fromLookup()` Método de.

Nombre	Tipo	Descripción
¿Implementar VPC?	`boolean`	Si se debe crear una VPC nueva basada en`vpcProps`en el que desplegar este patrón. Si se establece este valor`true`implementará la VPC mínima y más privada para ejecutar el patrón: • Una subred aislada en cada zona de disponibilidad utilizada por el programa CDK • `enableDnsHostnames` y`enableDns Support` ambos se establecerán en`true` Si esta propiedad es`true`, luego`existingVpc` No se puede especificar . El valor predeterminado es `false`.

Nombre	Tipo	Descripción
¿VPCProps?	`ec2.VpcProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas de la nueva VPC.`enableDns Hostnames` ,`enableDns Support` ,`natGateways` , `ysubnetConfiguratio n` están establecidos por el patrón, por lo que cualquier valor para las propiedad es suministradas aquí será anulado. Si`deployVpc` no es`true`, esta propiedad se pasará por alto.
QueueEnvironmentVa riableName?	`string`	Nombre opcional para el conjunto de variables de entorno URL de cola SQS para la función Lambda.

Patrón

Nombre	Tipo	Descripción
DeadLetterQueue?	`sqs.Queue`	Devuelve una instancia de la cola de letras muertas creada por el patrón, si se implementa una.
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.

Nombre	Tipo	Descripción
SQSqueue	`sqs.Queue`	Devuelve una instancia de la cola SQS creada por el patrón.
¿VPC?	`ec2.IVpc`	Devuelve una instancia de la VPC creada o utilizada por el patrón (si la hay). Puede ser una VPC creada por el patrón o una VPC suministrada al constructor del patrón.

Configuración predeterminada

La implementación inmediata del componente fijo sin anulación establecerá los siguientes valores predeterminados:

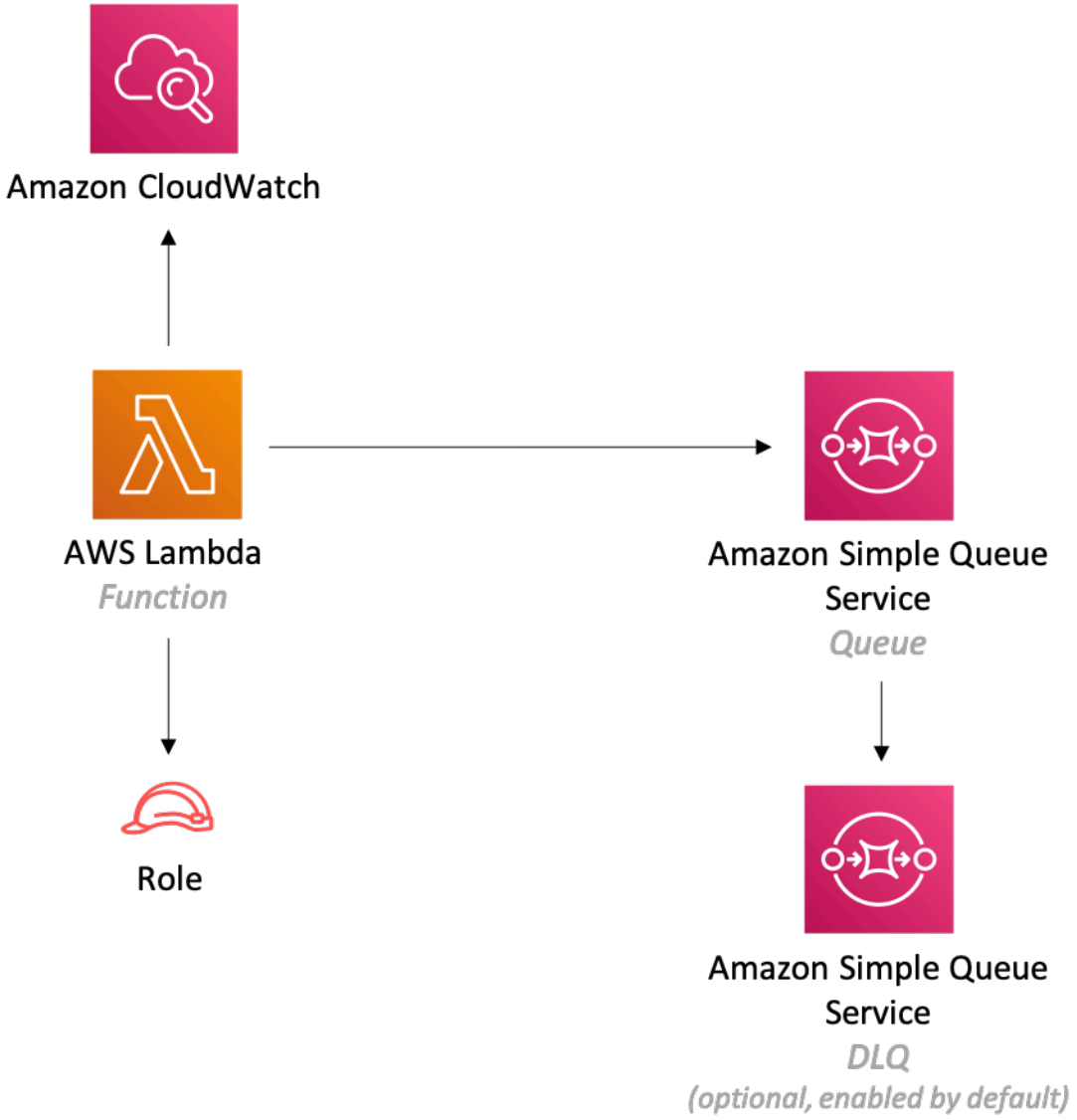
Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Permita que la función sólo envíe mensajes a la cola (la depuración se puede habilitar usando el comando`enableQueuePurge`).
- Habilite el X-Ray streo
- Configuración de las variables de entorno:
  - `SQS_QUEUE_URL`
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Cola de Amazon SQS

- Aplicación de la cola de mensajes fallidos de SQS para la cola de SQS de origen.
- Habilite el cifrado en el lado de servidor para la cola de SQS de origen mediante la clave KMS administrada de AWS.
- Aplicación del cifrado de los datos en tránsito.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-lambda-sqs

aws-lambda-sqs-lambda

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersionado semántico- El modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_lambda_sqs_lambda`
Mecanografía	`@aws-solutions-constructs/aws-lambda-sqs-lambda`
Java	`software.amazon.awsconstructs.services.lambdasqslambda`

Overview

Este patrón de soluciones de AWS implementa (1) una función de AWS Lambda configurada para enviar mensajes a una cola; (2) una cola de Amazon SQS; y (3) una función de AWS Lambda configurada para consumir mensajes de la cola.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { LambdaToSqsToLambda, LambdaToSqsToLambdaProps } from "@aws-solutions-constructs/aws-lambda-sqs-lambda";

new LambdaToSqsToLambda(this, 'LambdaToSqsToLambdaPattern', {
    producerLambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/producer-function/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda/producer-function`),
        handler: 'index.handler'
```

```
    },
    consumerLambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/consumer-function/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda/consumer-function`),
        handler: 'index.handler'
    }
});
```

Initializer

```
new LambdaToSqsToLambda(scope: Construct, id: string, props: LambdaToSqsToLambdaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`LambdaToSqsToLambdaProps`

Patrón de construcción

Nombre	Tipo	Descripción
ProductorLambdaobj existente Productor?	`lambda.Function`	Una función Lambda opcional existente que se utilizará en lugar de la función predeterminada para enviar mensajes a la cola. Proporcionando tanto esto como`producerLambdaFunctionProps` provocará un error.
ProductorLambdafunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades

Nombre	Tipo	Descripción
		predeterminadas de la función Lambda del productor.
ExistingQueueObj?	`sqs.Queue`	Una cola SQS opcional existente que se utilizará en lugar de la cola predeterm inada. Proporcionando tanto esto como`queueProp s` provocará un error.
¿Props en cola?	`sqs.QueueProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas de la cola SQS. Proporcionando tanto esto como`existingQ ueueObj` provocará un error.
DeployDeadLetterQueue?	`boolean`	Si se va a crear una cola secundaria para utilizars e como cola de mensajes fallidos. El valor predeterm inado es `true`.
DeadletterQueueProps?	`sqs.QueueProps`	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para la cola de letras muertas. Solo se usa si se usa el`deployDeadLetterQu eue` La propiedad se establece en`true`.

Nombre	Tipo	Descripción
¿maxReceiveCount?	`number`	El número de veces que un mensaje se puede quitar sin éxito antes de que se traslade a la cola de mensajes fallidos. El valor predeterminado es `15`.
ExistenteConsumerlambdaobj?	`lambda.Function`	Función Lambda opcional existente que se utilizará en lugar de la función predeterminada para recibir/consumir mensajes de la cola. Proporcionando tanto esto como`consumerLambdaFunctionProps` provocará un error.
ConsumerLambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda de consumo.
QueueEnvironmentVariableName?	`string`	Nombre opcional para el conjunto de variables de entorno URL de cola SQS para la función Lambda del productor.

Propiedades de patrón

Nombre	Tipo	Descripción
ConsumerLambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda de consumo creada por el patrón.
DeadLetterQueue?	`sqs.Queue`	Devuelve una instancia de la cola de letras muertas creada por el patrón, si se implementa una.
ProductorLambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda productor creada por el patrón.
SQSqueue	`sqs.Queue`	Devuelve una instancia de la cola SQS creada por el patrón.

Configuración predeterminada

La implementación lista para usar de este constructo (sin propiedades anuladas) se ajustará a los siguientes valores predeterminados:

Funciones AWS Lambda

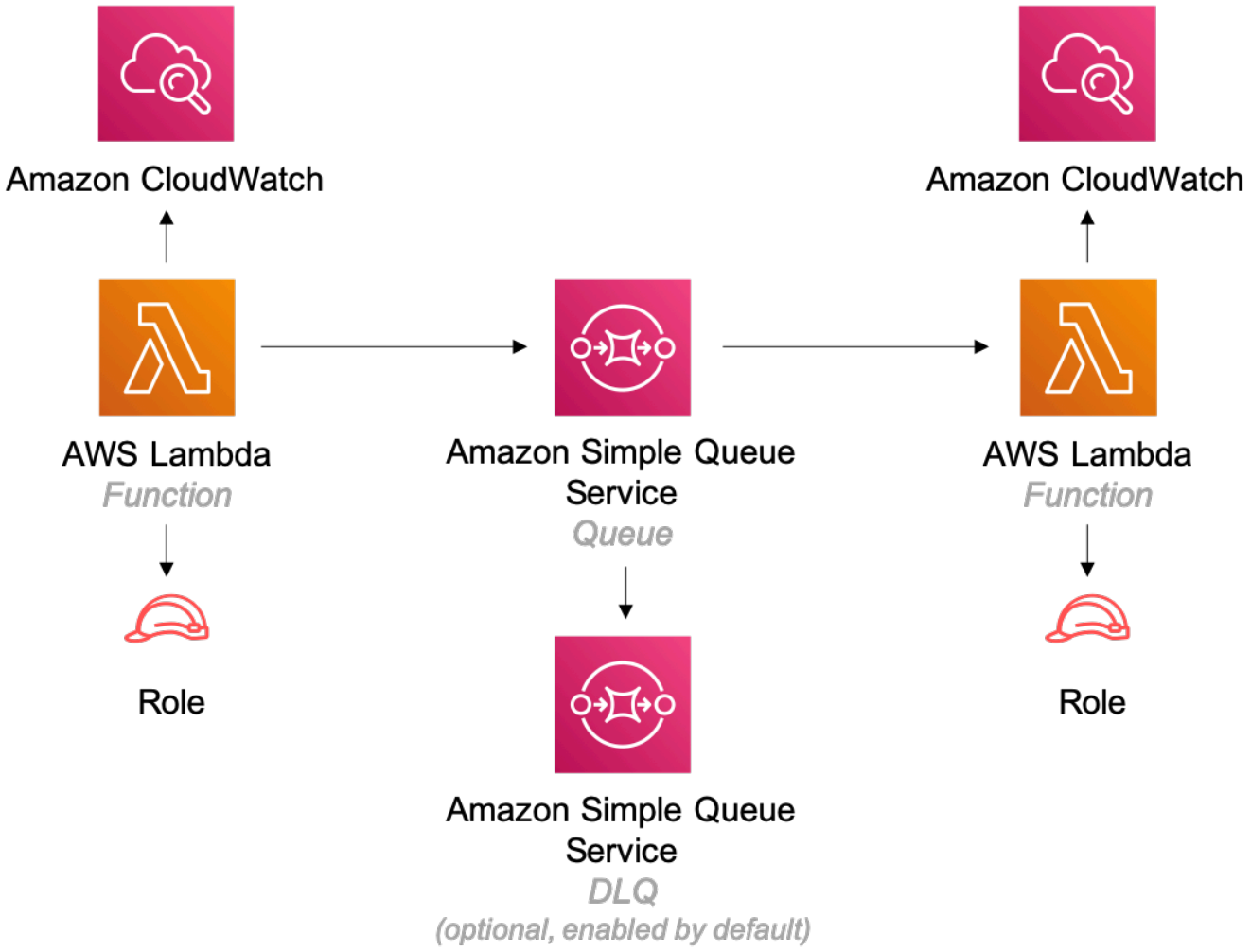
- Configurar el rol de IAM de acceso limitado con privilegios para las funciones de Lambda.
- Habilite la reutilización de conexiones con las funciones Keep-Alive para NodeJS Lambda.
- Habilite el X-Ray streo
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Cola de Amazon SQS

- Despliegue una cola de mensajes fallidos de la cola principal.

- Habilite el cifrado en el lado de servidor para la cola principal mediante una clave KMS gestionada de AWS.
- Aplicación del cifrado de los datos en tránsito

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:

	@aws -soluciones-constructs/aws-lambda-sqs-lambda

aws-lambda-step-función

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticas. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_lambda_step_function`
Mecanografía	`@aws-solutions-constructs/aws-lambda-step-function`
Java	`software.amazon.awsconstructs.services.lambdastepfunction`

Overview

Este constructo de soluciones de AWS implementa una función de AWS Lambda conectada a una función de pasos de AWS.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { LambdaToStepFunction } from '@aws-solutions-constructs/aws-lambda-step-function';
import * as stepfunctions from '@aws-cdk/aws-stepfunctions';

const startState = new stepfunctions.Pass(this, 'StartState');
```

```
new LambdaToStepFunction(this, 'LambdaToStepFunctionPattern', {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    },
    stateMachineProps: {
      definition: startState
    }
});
```

Initializer

```
new LambdaToStepFunction(scope: Construct, id: string, props:
 LambdaToStepFunctionProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`LambdaToStepFunctionProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario

Nombre	Tipo	Descripción
		para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si `unexistingLambdaObj` se encuentra.
StateMachineProps	`sfn.StateMachineProps`	El usuario proporcionó accesorios para SFN.State Machine.
CreateCloudWatchalarms	`boolean`	Si desea crear alarmas recomendadas de CloudWatch.
LogGroupProps?	`logs.LogGroupProps`	Los accesorios opcionales proporcionados por el usuario para reemplazar los accesorios predeterminados para el grupo de registros de CloudWatch Logs.
StateMachineEnvironmentVariableName	`string`	Nombre opcional para el conjunto de variables de entorno de máquina de estado de Step Functions para la función Lambda del productor.

Propiedades de patrón

Nombre	Tipo	Descripción
¿CloudWatcharms?	`cloudwatch.Alarm[]`	Devuelve una lista de una o más alarmas de CloudWatch creadas por el patrón.

Nombre	Tipo	Descripción
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.
StateMachine	`sfn.StateMachine`	Devuelve una instancia de la máquina de estado creada por el patrón.
StateMachineLogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por el patrón para la máquina de estado.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

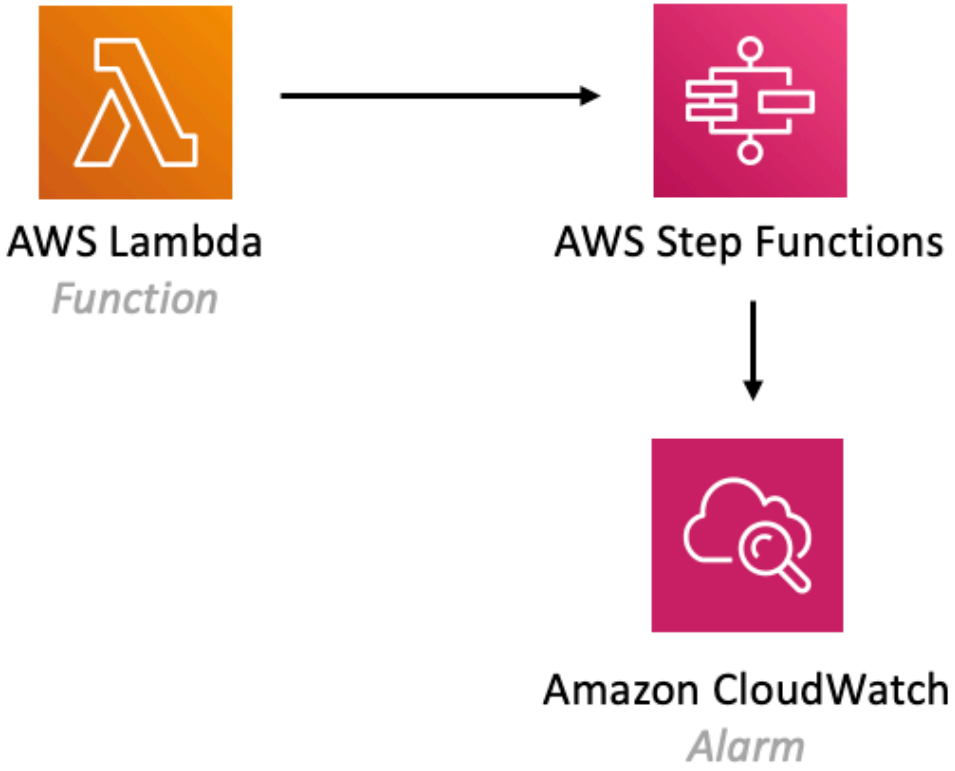
Función de AWS Lambda

- Configure un rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con las funciones Keep-Alive para NodeJS Lambda.
- Active el rastreo de X-Ray.
- Configuración de las variables de entorno:
  - `STATE_MACHINE_ARN` (predeterminado)
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Máquina de estado de AWS Step Functions

- Implemente alarmas de CloudWatch de prácticas recomendadas para la máquina de estado de AWS Step Functions.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-lambda-step-function

aws-s3-lambda

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_s3_lambda`
Mecanografía	`@aws-solutions-constructs/aws-s3-lambda`
Java	`software.amazon.awsconstructs.services.s3lambda`

Overview

Este constructo de soluciones de AWS implementa un depósito de Amazon S3 conectado a una función de AWS Lambda.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { S3ToLambdaProps, S3ToLambda } from '@aws-solutions-constructs/aws-s3-lambda';

new S3ToLambda(this, 'test-s3-lambda', {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    },
```

```
});
```

Initializer

```
new S3ToLambda(scope: Construct, id: string, props: S3ToLambdaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`S3ToLambdaProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si un`existingLambdaObj` se encuentra.
¿ExistenteBuckeTobj?	`s3.Bucket`	Instancia existente del objeto S3 Bucket. Si esto se proporciona,

Nombre	Tipo	Descripción
		entonces también proporcio nabucketProps es un error.
BucketProps?	s3.BucketProps	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas del depósito. Se pasa por alto si unexistingBucketObj se encuentra.
S3EventSourceProps?	S3EventSourceProps	Los accesorios proporcio nados por el usuario opcionale s para anular los accesorio s predeterminados para S3EventSourceProps

Propiedades de patrón

Nombre	Tipo	Descripción
LambdaFunction	lambda.Function	Devuelve una instancia de la función Lambda creada por el patrón.
¿S3 Bucket?	s3.Bucket	Devuelve una instancia del bucket S3 creado por el patrón.
¿S3LoggingBucket?	s3.Bucket	Devuelve una instancia del depósito de registro creado por el patrón para el bucket S3.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

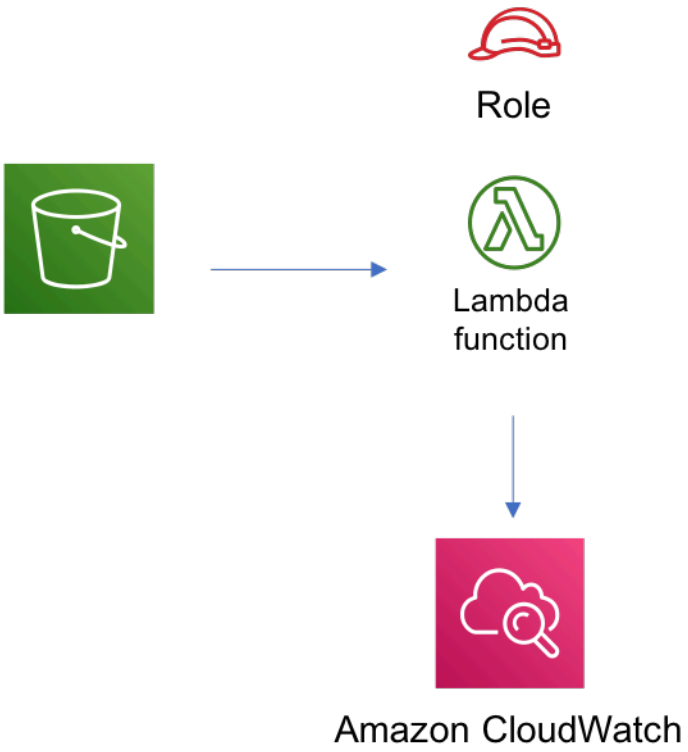
Bucket de Amazon S3

- Configurar el registro de acceso para S3 Bucket.
- Habilite el cifrado del lado del servidor para S3 Bucket mediante la clave KMS administrada por AWS.
- Active el control de versiones para S3 Bucket.
- No permita el acceso público para S3 Bucket.
- Mantenga el Bucket S3 al eliminar la pila CloudFormation.
- Aplicación del cifrado de los datos en tránsito.
- Aplica una regla de ciclo de vida para mover versiones de objetos no actuales al almacenamiento de Glacier después de 90 días.

Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Active el rastreo de X-Ray.
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-s3-lambda

aws-s3-sqs

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_s3_sqs`
Mecanografiado	`@aws-solutions-constructs/aws-s3-sqs`
Java	`software.amazon.awsconstructs.services.s3sqs`

Overview

Este constructo de soluciones de AWS implementa un depósito de Amazon S3 configurado para enviar notificaciones a una cola de Amazon SQS.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { S3ToSqs } from "@aws-solutions-constructs/aws-s3-sqs";

new S3ToSqs(stack, 'S3ToSQSPattern', {});
```

Initializer

```
new S3ToSqs(scope: Construct, id: string, props: S3ToSqsProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`S3ToSqsProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteBuckeTobj?	`s3.Bucket`	Instancia existente del objeto S3 Bucket. Si esto se proporciona, entonces también proporcio nabucketProps es un error.
¿BucketProps?	`s3.BucketProps`	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para el bucket S3.
s3EventTypes?	`s3.EventType[]`	Los tipos de eventos S3 que activarán la notificación. El valor predeterminado es `s3.EventType.OBJEC T_CREATED` .
S3EventFilters?	`s3.NotificationKey Filter[]`	Las reglas de filtro de clave de objeto S3 para determina r qué objetos activan este evento. Si no se especifica, no se aplicará ninguna regla de filtro.
ExistingQueueObj?	`sqs.Queue`	Una cola SQS opcional existente que se utilizará en lugar de la cola predeterm inada. Proporcionando tanto esto comoqueueProp s provocará un error. Si la cola SQS está cifrada, la clave KMS utilizada para el cifrado

Nombre	Tipo	Descripción
		debe ser un CMK administr ado por el cliente.
¿Props en cola?	`sqs.QueueProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas de la cola SQS. Se pasa por alto si `unexistingQueueObj` se encuentra.
DeadletterQueueProps?	`sqs.QueueProps`	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para la cola de letras muertas. Solo se usa si el`deployDeadLetterQu eue` se establece en verdadero.
DeployDeadLetterQueue?	`boolean`	Si se crea una cola secundari a para utilizarse como cola de mensajes fallidos. El valor predeterminado es `true`.
¿maxReceiveCount?	`number`	El número de veces que un mensaje se puede quitar sin éxito antes de que se traslade a la cola de mensajes fallidos. El valor predeterminado es `15`.

Nombre	Tipo	Descripción
EnableEncryptionWithCustomerManagedKey?	`boolean`	Si desea utilizar una clave KMS, ya sea administrada por esta aplicación CDK o importada. Si se importa una clave de cifrado, debe especificarse en el campo`encryptionKey` para esta construcción.
encryptionKey	`kms.Key`	Una clave de cifrado opcional existente que se utilizará en lugar de la clave de cifrado predeterminada.
EncryptionKeyProps?	`kms.KeyProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la clave de cifrado.

Propiedades de patrón

Nombre	Tipo	Descripción
SQSqueue	`sqs.Queue`	Devuelve una instancia de la cola SQS creada por el patrón.
DeadLetterQueue?	`sqs.Queue`	Devuelve una instancia de la cola de letras muertas creada por el patrón, si se implementa una.

Nombre	Tipo	Descripción
encryptionKey	`kms.IKey`	Devuelve una instancia de la clave de cifrado creada por el patrón.
¿S3 Bucket?	`s3.Bucket`	Devuelve una instancia del bucket S3 creado por el patrón.
¿S3LoggingBucket?	`s3.Bucket`	Devuelve una instancia del depósito de registro creado por el patrón para el bucket S3.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Bucket de Amazon S3

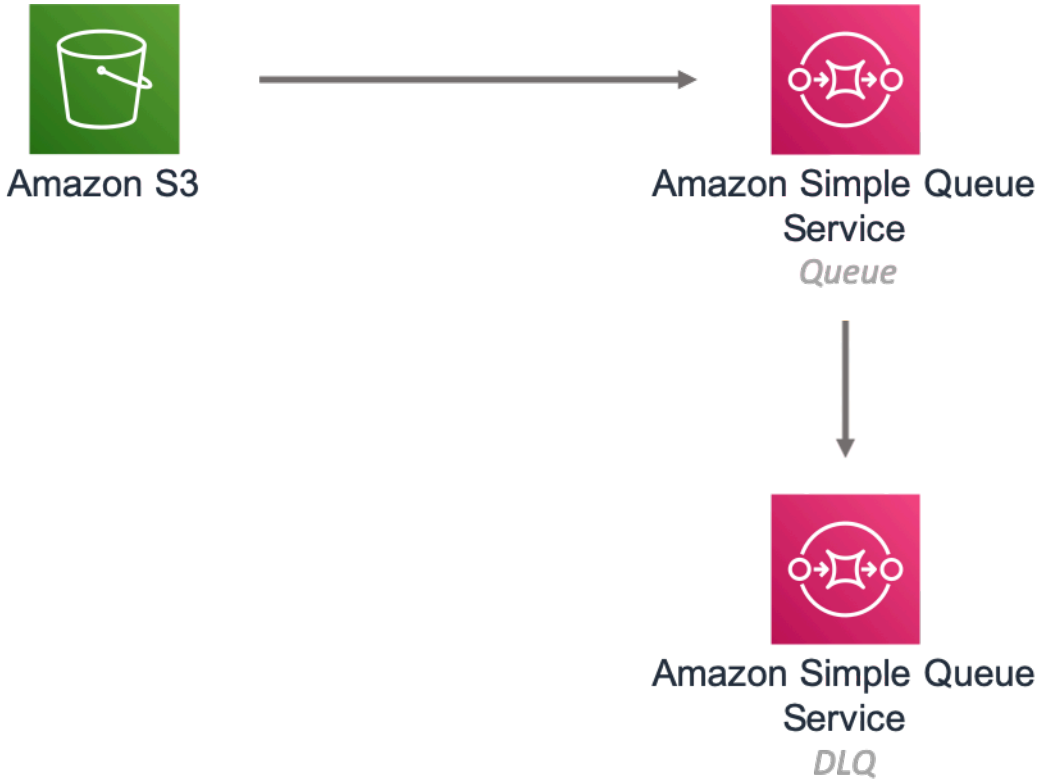
- Configurar el registro de acceso para S3 Bucket
- Habilitar el cifrado del lado del servidor para S3 Bucket utilizando la clave KMS administrada por AWS
- Activar el control de versiones para S3 Bucket
- No permitir el acceso público para S3 Bucket
- Conservar el depósito de S3 al eliminar la pila de CloudFormation
- Aplicación del cifrado de los datos en tránsito
- Aplica una regla de ciclo de vida para mover versiones de objetos no actuales al almacenamiento de Glacier después de 90 días

Cola de Amazon SQS

- Configurar permisos de acceso de privilegios mínimos para SQS Queue
- Implementar cola de mensajes fallidos de SQS para la cola de origen SQS

- Habilite el cifrado en el lado de servidor para la cola de SQS mediante la clave KMS administrada por el cliente
- Aplicación del cifrado de los datos en tránsito

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-s3-sqs

función de paso aws-s3

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones

semánticasEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_s3_step_function`
TypeScript	`@aws-solutions-constructs/aws-s3-step-function`
Java	`software.amazon.awsconstructs.services.s3stepfunction`

Overview

Este constructo de soluciones de AWS implementa un depósito de Amazon S3 conectado a una función de pasos de AWS.

> **Note**
>
> Esta construcción utiliza Amazon EventBridge (Amazon CloudWatch Events) para activar AWS Step Functions. EventBridge es más flexible, pero activar Step Functions con notificaciones de eventos de S3 tiene menos latencia y es más rentable. Si el costo y/o la latencia son un problema, debe considerar implementar`aws-s3-lambda`y`aws-lambda-stepfunctions`en lugar de este constructo.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { S3ToStepFunction, S3ToStepFunctionProps } from '@aws-solutions-constructs/aws-s3-step-function';
```

```
import * as stepfunctions from '@aws-cdk/aws-stepfunctions';

const startState = new stepfunctions.Pass(this, 'StartState');

new S3ToStepFunction(this, 'test-s3-step-function-stack', {
    stateMachineProps: {
      definition: startState
    }
});
```

Initializer

```
new S3ToStepFunction(scope: Construct, id: string, props: S3ToStepFunctionProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`S3ToStepFunctionProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteBuckeTobj?	`s3.IBucket`	Instancia existente del objeto S3 Bucket. Si se proporcio na esto, entonces también proporcionabucketProps es un error.
¿BucketProps?	`s3.BucketProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas del depósito. Se pasa por alto si

Nombre	Tipo	Descripción
		`unexistingBucketObj` se encuentra.
StateMachineProps	`sfn.StateMachineProps`	El usuario ha proporcionado accesorios opcionales para anular los accesorios predeterminados para SFN.StateMachine.
¿EventRuleProps?	`events.RuleProps`	El usuario opcional proporcionó EventRuleProps para anular los valores predeterminados.
Implementar CloudTrail?	`boolean`	Si desea implementar un Trail en AWS CloudTrail para registrar eventos de API en Amazon S3. El valor predeterminado es `true`.
CreateCloudWatchalarms	`boolean`	Si desea crear alarmas recomendadas de CloudWatch.
LogGroupProps?	`logs.LogGroupProps`	Concede opcionales proporcionados por el usuario para anular los accesorios predeterminados para el grupo de registros de CloudWatch Logs.

Propiedades de patrón

Nombre	Tipo	Descripción
¿Trail de nubes?	`cloudtrail.Trail`	Devuelve una instancia de la pista de Cloudtrail creada por el patrón.
¿CloudTrailBucket?	`s3.Bucket`	Devuelve una instancia del bucket creado por el patrón para almacenar datos de rastreo de Cloudtrail.
CloudTrailLogingBucket?	`s3.Bucket`	Devuelve una instancia del depósito de registro creado por el patrón para el depósito principal utilizado por la pista de Cloudtrail.
¿CloudWatchalarms?	`cloudwatch.Alarm[]`	Devuelve una lista de una o más alarmas de CloudWatch creadas por el patrón.
¿S3 Bucket?	`s3.Bucket`	Devuelve una instancia del bucket S3 creado por el patrón.
¿S3LoggingBucket?	`s3.Bucket`	Devuelve una instancia del depósito de registro creado por el patrón para el bucket S3.
StateMachine	`sfn.StateMachine`	Devuelve una instancia de la máquina de estado creada por el patrón.
StateMachineLogGroup	`logs.LogGroup`	Devuelve una instancia del grupo de registros creado por

Nombre	Tipo	Descripción
		el patrón para la máquina de estado.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

Bucket de Amazon S3

- Configurar el registro de acceso para S3 Bucket.
- Habilite el cifrado del lado del servidor para S3 Bucket mediante la clave KMS administrada por AWS.
- Active el control de versiones para S3 Bucket.
- No permita el acceso público para S3 Bucket.
- Mantenga el Bucket S3 cuando elimine la pila CloudFormation.
- Aplicación del cifrado de los datos en tránsito.
- Aplica una regla de ciclo de vida para mover versiones de objetos no actuales al almacenamiento de Glacier después de 90 días.

AWS CloudTrail

- Configure un Trail en AWS CloudTrail para registrar eventos de API en Amazon S3 relacionados con el Bucket creado por el Construct.

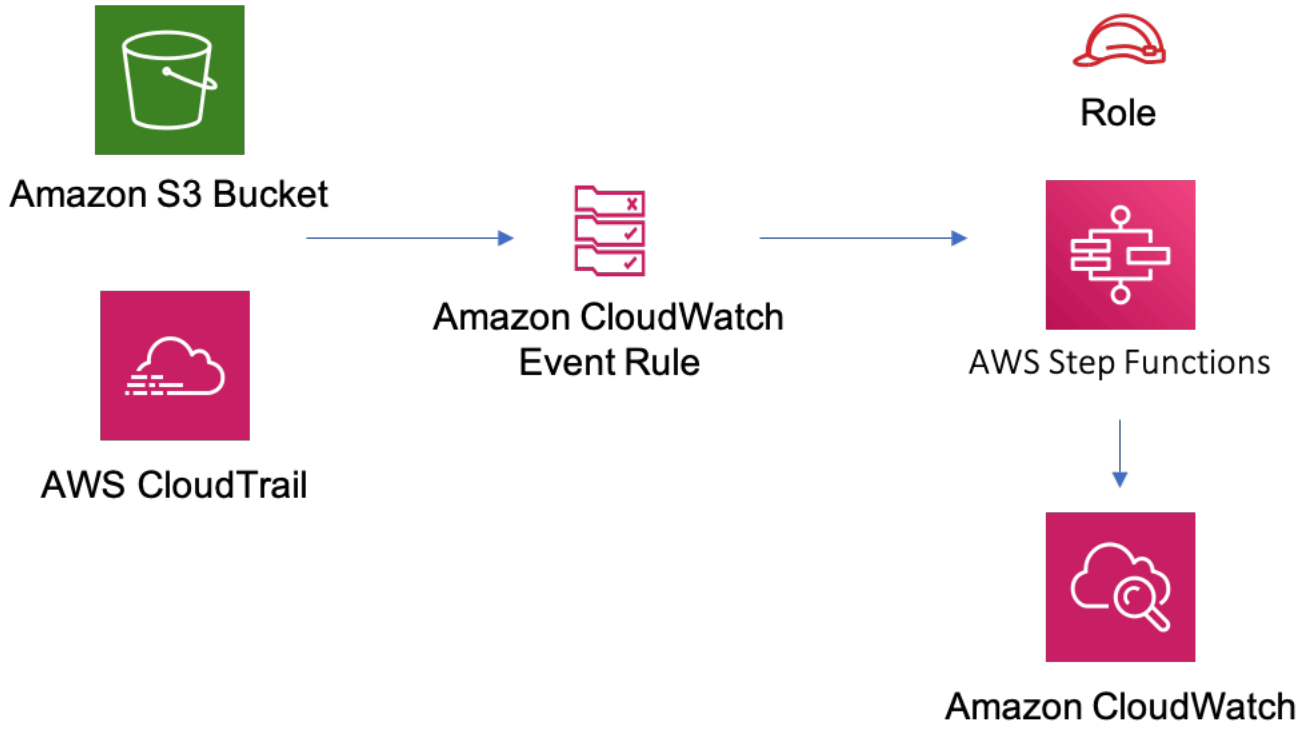
Reglas de Amazon CloudWatch Events

- Conceda permisos de privilegios mínimos a CloudWatch Events para activar la función Lambda.

AWS Step Function

- Active el registro de CloudWatch para API Gateway.
- Implemente las prácticas recomendadas de CloudWatch Alarms para la función Step.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -solutions-constructs/aws-s3-step-function

aws-sns-lambda

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasmodelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_sns_lambda`
Mecanografía	`@aws-solutions-constructs/aws-sns-lambda`
Java	`software.amazon.awsconstructs.services.snslambda`

Overview

Este constructo de soluciones de AWS implementa un Amazon SNS conectado a una función de AWS Lambda.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { SnsToLambda, SnsToLambdaProps } from "@aws-solutions-constructs/aws-sns-
lambda";

new SnsToLambda(this, 'test-sns-lambda', {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    }
});
```

Initializer

```
new SnsToLambda(scope: Construct, id: string, props: SnsToLambdaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`SnsToLambdaProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si se`existingLambdaObj` se encuentra.
¿ExistentetoPicobj?	`sns.Topic`	Instancia existente del objeto SNS Topic, proporcionando tanto esto como`topicProps` provocará un error.
¿TopicProps?	`sns.TopicProps`	Propiedades proporcionadas por el usuario opcionales para anular las propiedades predeterminadas para el tema SNS.

Propiedades de patrón

Nombre	Tipo	Descripción
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.
snsTopic	`sns.Topic`	Devuelve una instancia del tema SNS creado por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

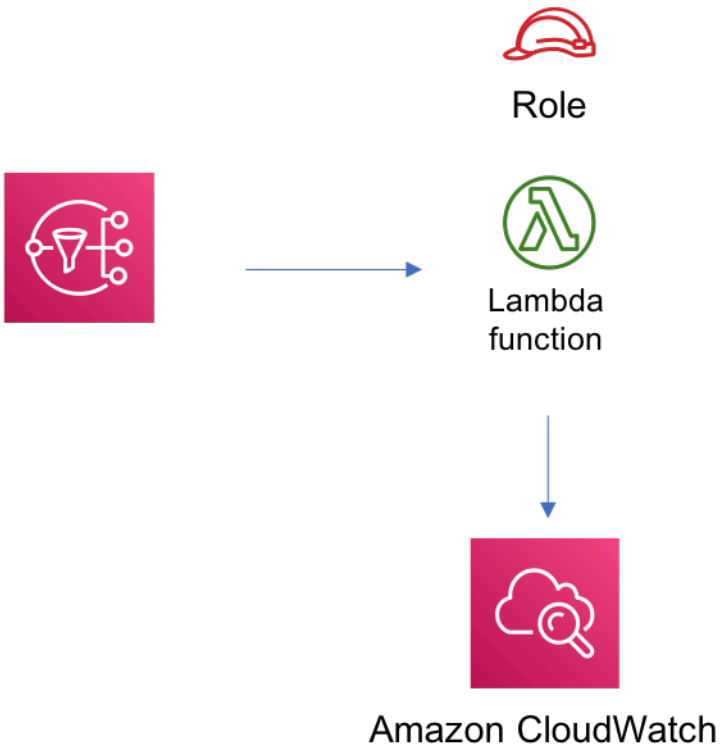
Tema de Amazon SNS

- Configurar permisos de acceso de privilegios mínimos para el tema SNS.
- Habilite el cifrado del lado del servidor mediante la clave KMS administrada por AWS.
- Aplicación del cifrado de los datos en tránsito.

Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Active el rastreo de X-Ray.
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-sns-lambda

aws-sns-sqs

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersionado semánticoEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_sns_sqs`
Mecanografiado	`@aws-solutions-constructs/aws-sns-sqs`
Java	`software.amazon.awsconstructs.services.snssqs`

Overview

Este construcción de soluciones de AWS implementa un tema de Amazon SNS conectado a una cola de Amazon SQS.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
import { SnsToSqs, SnsToSqsProps } from "@aws-solutions-constructs/aws-sns-sqs";
import * as iam from '@aws-cdk/aws-iam';

const snsToSqsStack = new SnsToSqs(this, 'SnsToSqsPattern', {});

// Grant yourself permissions to use the Customer Managed KMS Key
const policyStatement = new iam.PolicyStatement({
    actions: ["kms:Encrypt", "kms:Decrypt"],
    effect: iam.Effect.ALLOW,
    principals: [ new iam.AccountRootPrincipal() ],
    resources: [ "*" ]
});

snsToSqsStack.encryptionKey?.addToResourcePolicy(policyStatement);
```

Initializer

```
new SnsToSqs(scope: Construct, id: string, props: SnsToSqsProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`SnsToSqsProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistentetoPicobj?	`sns.Topic`	Instancia existente del objeto SNS Topic, proporcionando tanto esto como`topicProps` provocará un error.
¿TopicProps?	`sns.TopicProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas del tema SNS. Se pasa por alto si `unexistingTopicObj` se encuentra.
ExistingQueueObj?	`sqs.Queue`	Una cola SQS opcional existente que se utilizará en lugar de la cola predeterminada. Proporcionando tanto esto como`queueProps` provocará un error.

Nombre	Tipo	Descripción
¿Props en cola?	`sqs.QueueProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedad es predeterminadas de la cola SQS. Se pasa por alto si `unexistingQueueObj` se encuentra.
DeployDeadLetterQueue?	`boolean`	Si se crea una cola de secundaria para utilizars e como cola de mensajes fallidos. El valor predeterm inado es `true`.
DeadletterQueueProps?	`sqs.QueueProps`	Props opcionales proporcio nados por el usuario para anular los apoyos predeterm inados para la cola de letras muertas. Solo se usa si el`deployDeadLetterQu eue` se establece en true.
¿maxReceiveCount?	`number`	El número de veces que un mensaje se puede quitar sin éxito antes de que se traslade a la cola de mensajes fallidos. El valor predeterminado es `15`.

Nombre	Tipo	Descripción
EnableEncryptionWithCustomerManagedKey?	`boolean`	Si desea utilizar una clave de cifrado administrada por el cliente, ya sea administrada por esta aplicación CDK o importada. Si importa una clave de cifrado, debe especificarse en el campo `encryptionKey` para esta construcción.
encryptionKey	`kms.Key`	Una clave de cifrado opcional existente que se utilizará en lugar de la clave de cifrado predeterminada.
EncryptionKeyProps?	`kms.KeyProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la clave de cifrado.

Propiedades de patrón

Nombre	Tipo	Descripción
SNSTopic	`sns.Topic`	Devuelve una instancia del tema SNS creado por el patrón.
encryptionKey	`kms.Key`	Devuelve una instancia de la clave de cifrado creada por el patrón.

Nombre	Tipo	Descripción
SQSqueue	`sqs.Queue`	Devuelve una instancia de la cola SQS creada por el patrón.
DeadLetterQueue?	`sqs.Queue`	Devuelve una instancia de la cola de letras muertas creada por el patrón, si se implementa una.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

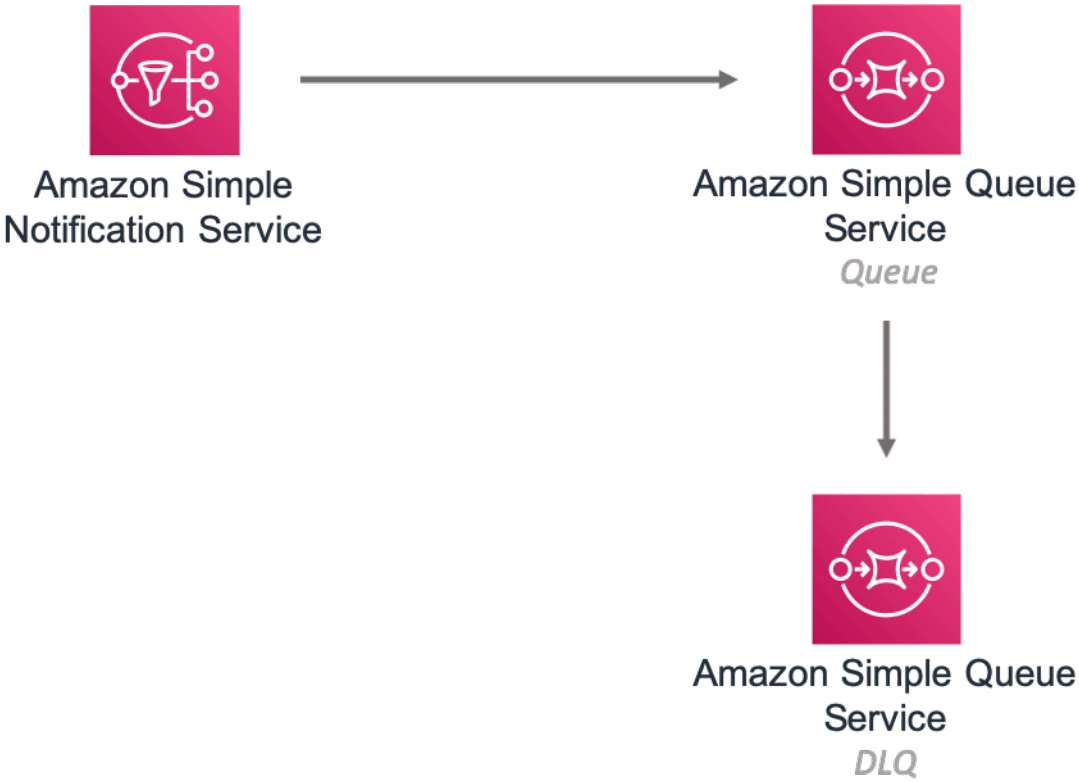
Tema de Amazon SNS

- Configurar permisos de acceso de privilegios mínimos para el tema SNS.
- Habilite el cifrado del lado del servidor mediante la clave KMS administrada por AWS.
- Aplicación del cifrado de los datos en tránsito.

Cola de Amazon SQS

- Configurar permisos de acceso de privilegios mínimos para la cola de SQS.
- Implemente la cola de mensajes fallidos para la cola de SQS de origen.
- Habilite el cifrado en el lado de servidor para la cola de SQS mediante la clave KMS administrada por el cliente.
- Aplicación del cifrado de los datos en tránsito.

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-sns-sqs

aws-sqs-lambda

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersionado semánticoEl modelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

Nota: Para garantizar una funcionalidad adecuada, los paquetes de AWS Solutions Constructs y los paquetes CDK de AWS del proyecto deben tener la misma versión.

Idioma	Paquete
Python	`aws_solutions_constructs.aws_sqs_lambda`
Mecanografía	`@aws-solutions-constructs/aws-sqs-lambda`
Java	`software.amazon.awsconstructs.services.sqslambda`

Overview

Este constructo de soluciones de AWS implementa una cola de Amazon SQS conectada a una función de AWS Lambda.

Aquí hay una definición mínima de patrón implementable en TypeScript:

```
const { SqsToLambda } = require('@aws-solutions-constructs/aws-sqs-lambda');

new SqsToLambda(stack, 'SqsToLambdaPattern', {
    lambdaFunctionProps: {
        runtime: lambda.Runtime.NODEJS_14_X,
        // This assumes a handler function in lib/lambda/index.js
        code: lambda.Code.fromAsset(`${__dirname}/lambda`),
        handler: 'index.handler'
    }
});
```

Initializer

```
new SqsToLambda(scope: Construct, id: string, props: SqsToLambdaProps);
```

Parámetros

- scope`Construct`
- id`string`
- props`SqsToLambdaProps`

Patrón de construcción

Nombre	Tipo	Descripción
¿ExistenteLambdaobj?	`lambda.Function`	Instancia existente del objeto Lambda Function, proporcionando tanto esto como`lambdaFunctionProps` provocará un error.
¿LambdaFunctionProps?	`lambda.FunctionProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la función Lambda. Se pasa por alto si `unexistingLambdaObj` se encuentra.
ExistingQueueObj?	`sqs.Queue`	Una cola SQS opcional existente que se utilizará en lugar de la cola predeterminada. Proporcionando tanto esto como`queueProps` provocará un error.
¿Props en cola?	`sqs.QueueProps`	Propiedades opcionales proporcionadas por el usuario para anular las propiedades predeterminadas de la cola SQS. Se pasa por alto si

Nombre	Tipo	Descripción
		`unexistingQueueObj` se encuentra.
DeployDeadLetterQueue?	`boolean`	Si se crea una cola secundaria para utilizarse como cola de mensajes fallidos. El valor predeterminado es `true`.
DeadletterQueueProps?	`sqs.QueueProps`	Props opcionales proporcionados por el usuario para anular los apoyos predeterminados para la cola de letras muertas. Solo se usa si el`deployDeadLetterQueue` se establece en verdadero.
¿maxReceiveCount?	`number`	El número de veces que un mensaje se puede quitar sin éxito antes de que se traslade a la cola de mensajes fallidos. El valor predeterminado es `15`.

Propiedades de patrón

Nombre	Tipo	Descripción
DeadLetterQueue?	`sqs.Queue`	Devuelve una instancia de la cola de letras muertas creada por el patrón, si se implementa una.
LambdaFunction	`lambda.Function`	Devuelve una instancia de la función Lambda creada por el patrón.

Nombre	Tipo	Descripción
SQSqueue	`sqs.Queue`	Devuelve una instancia de la cola SQS creada por el patrón.

Configuración predeterminada

La implementación lista para usar de este patrón sin anulación establecerá los siguientes valores predeterminados:

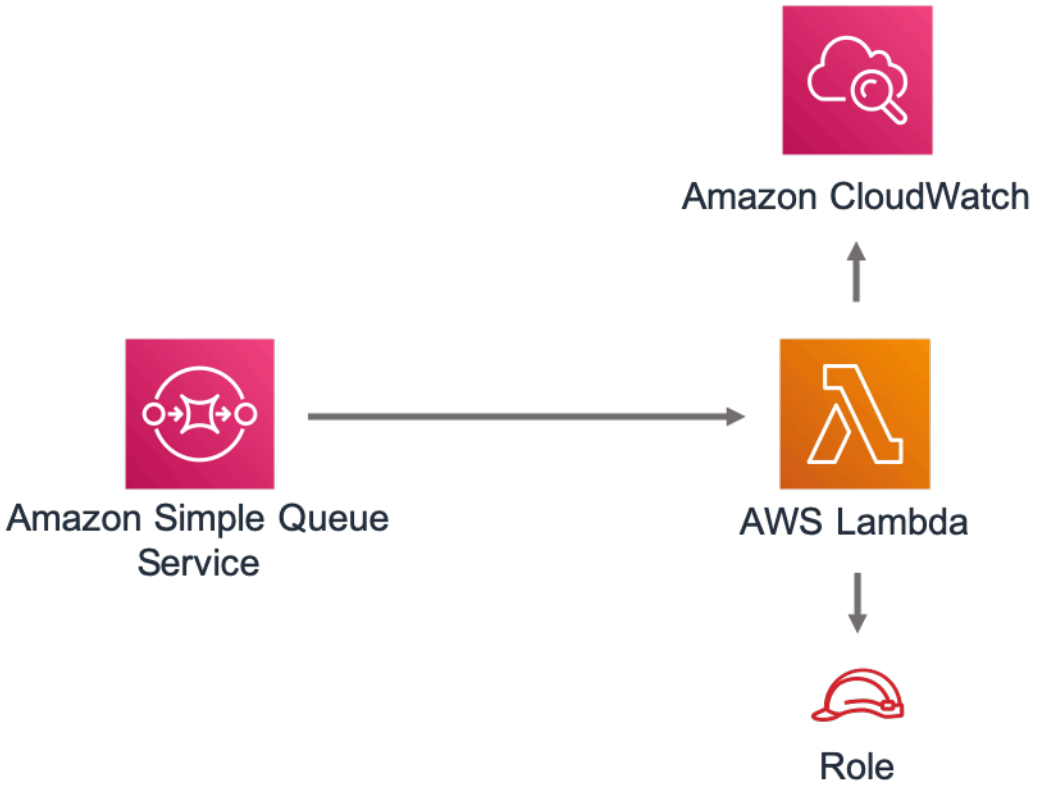
Cola de Amazon SQS

- Implemente la cola de mensajes fallidos de SQS para la cola de SQS de origen.
- Habilite el cifrado en el lado de servidor para la cola de SQS de origen mediante la clave KMS administrada de AWS.
- Cumplir el cifrado de los datos en tránsito.

Función de AWS Lambda

- Configurar el rol de IAM de acceso limitado con privilegios para la función Lambda.
- Habilite la reutilización de conexiones con la función Keep-Alive para NodeJS Lambda.
- Active el rastreo de X-Ray.
- Configuración de las variables de entorno:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(para Node 10.x y funciones superiores)

Architecture



GitHub

Para ver el código de este patrón, crear/ver problemas y solicitudes de extracción, y mucho más:	
	@aws -soluciones-constructs/aws-sqs-lambda

core

STABILITY EXPERIMENTAL

Todas las clases están en desarrollo activo y están sujetas a cambios o eliminación no compatibles con versiones anteriores en cualquier versión futura. Estos no están sujetos a laVersiones semánticasModelo. Esto significa que, aunque puede usarlos, es posible que necesite actualizar el código fuente cuando actualice a una versión más reciente de este paquete.

La biblioteca principal incluye los componentes básicos de AWS Solutions Comstructs. Define las clases principales que se utilizan en el resto de construcciones de soluciones de AWS.

Propiedades predeterminadas para construcciones CDK de AWS

La biblioteca principal establece las propiedades predeterminadas de los constructos CDK de AWS utilizados por los componentes fijos de soluciones de AWS.

Por ejemplo, el siguiente es el fragmento de propiedades predeterminadas para la construcción de S3 Bucket creada por AWS Solutions Constructs. De forma predeterminada, activará el cifrado del lado del servidor, el control de versiones del bucket, bloqueará todo el acceso público y configurará el registro de acceso S3.

```
{
  encryption: s3.BucketEncryption.S3_MANAGED,
  versioned: true,
  blockPublicAccess: s3.BlockPublicAccess.BLOCK_ALL,
  removalPolicy: RemovalPolicy.RETAIN,
  serverAccessLogsBucket: loggingBucket
}
```

Anular las propiedades predeterminadas

Las propiedades predeterminadas establecidas por la biblioteca Core pueden ser anuladas por las propiedades proporcionadas por el usuario. Por ejemplo, el usuario puede anular la propiedad Amazon S3 Block Public Access para cumplir requisitos específicos.

```
const stack = new cdk.Stack();

const props: CloudFrontToS3Props = {
  bucketProps: {
    blockPublicAccess: {
      blockPublicAcls: false,
      blockPublicPolicy: true,
      ignorePublicAcls: false,
      restrictPublicBuckets: true
    }
  }
};
```

```
new CloudFrontToS3(stack, 'test-cloudfront-s3', props);

expect(stack).toHaveResource("AWS::S3::Bucket", {
  PublicAccessBlockConfiguration: {
    BlockPublicAcls: false,
    BlockPublicPolicy: true,
    IgnorePublicAcls: false,
    RestrictPublicBuckets: true
  },
});
```

Advertencias de reemplazo de propiedades

Cuando una propiedad proporcionada por el usuario reemplaza una propiedad predeterminada de la biblioteca Core, los componentes fijos emitirán uno o más mensajes de advertencia a la consola en los que se resaltan los cambios. Estos mensajes están destinados a proporcionar conocimiento de la situación al usuario y evitar anulaciones involuntarias que podrían crear riesgos de seguridad. Estos mensajes aparecerán siempre que se ejecuten comandos relacionados con la implementación/compilación, incluyendo`cdk deploy`,`cdk synth`,`npm test`, etc.:

Mensaje de ejemplo:`AWS_CONSTRUCTS_WARNING: An override has been provided for the property: BillingMode. Default value: 'PAY_PER_REQUEST'. You provided: 'PROVISIONED'.`

Alternar advertencias de anulación

Los mensajes de advertencia de anulación están habilitados de forma predeterminada, pero se pueden activar o desactivar explícitamente mediante la opción`overrideWarningsEnabled`Variable de shell.

- Para explícitamenteDesactivación deadvertencias de anulación, ejecutar`export overrideWarningsEnabled=false`.
- Para explícitamenteActivaradvertencias de anulación, ejecutar`export overrideWarningsEnabled=true`.
- Para volver al valor predeterminado, ejecuteun`set overrideWarningsEnabled`.

Revisiones del documento

Si desea recibir notificaciones sobre las actualizaciones de AWS Solutions Constructs, suscríbase a la fuente RSS.

update-history-change	update-history-description	update-history-date
Contenido actualizado	Se ha añadido el patrón aws-lambda-ssmstringparameter. Otras actualizaciones de contenido menores.	27 de mayo de 2021
Contenido actualizado	Añadido patrón aws-lambda-secretsmanager. Otras actualizaciones de contenido menores.	12 de mayo de 2021
Contenido actualizado	Actualizaciones de propiedades para seleccionar patrones *-lambda. Otras actualizaciones de contenido menores.	17 de abril de 2021
Contenido actualizado	Se ha corregido un problema en el Tutorial para usuarios de Python y se han actualizado ejemplos de propiedades para construcciones que contienen funciones Lambda.	30 de marzo de 2021
Contenido actualizado	Correcciones/actualizaciones menores a los apoyos de patrón y los ajustes predeterminados para los patrones seleccionados.	8 de marzo de 2021

Contenido actualizado	Correcciones/actualizaciones menores para el contenido del tutorial.	4 de marzo de 2021
Contenido actualizado	Se ha añadido`aws-lambda-sagemakerendpoint` y propiedades actualizadas para los patrones de Kinesis Firehose seleccionados.	24 de febrero de 2021
Contenido actualizado	Se ha añadido`aws-kinesisstreams-gluejob` y pasos de tutorial actualizados para los usuarios de Python.	17 de febrero de 2021
Contenido actualizado	Propiedades actualizadas para`aws-cloudfront-*`Patrones.	9 de febrero de 2021
Contenido actualizado	Se agregó un enlace a GitHub para cada patrón.	5 de febrero de 2021
Contenido actualizado	Se actualizaron las propiedades de los patrones seleccionados.	1 de febrero de 2021
Contenido actualizado	Se actualizó la documentación de las propiedades y la configuración predeterminada de los patrones de selección.	4 de enero de 2021
Contenido actualizado	Se han añadido nuevos patrones: aws-cloudfront-mediastore y aws-s3-sqs.	20 de diciembre de 2020
Contenido actualizado	Se ha eliminado el patrón aws-lambda-sagemaker.	17 de noviembre de 2020

Contenido actualizado	Se han añadido nuevos patrones: aws-events-rule-kinesisstreams, aws-events-rule-kinesisfirehose-s3 y aws-lambda-sagemaker.	27 de octubre de 2020
Contenido actualizado	Actualizado para reflejar el cambio de ruptura en aws-events-rule-sns y aws-events-rule-sqs patrones: nombres de clase e interfaz cambiaron a pascal case.	22 de octubre de 2020
Contenido actualizado	Se han añadido patrones aws-apigateway-sagemakerend point y aws-kinesisstreams -kinesisfirehose-s3; otras actualizaciones menores al contenido existente.	20 de octubre de 2020
Contenido actualizado	Se agregó el patrón aws-apigateway-iot; otras actualiza ciones menores al contenido existente.	7 de octubre de 2020
Contenido actualizado	Se actualizaron los fragmentos de código de patrón implement ables mínimos y los valores predeterminados de las prácticas recomendadas para todos los patrones.	5 de octubre de 2020
Contenido actualizado	Se actualizaron las propiedad es del patrón aws-kines isstreams-lambda para reflejar el cambio de ruptura.	14 de septiembre de 2020

Contenido actualizado	Corrección menor a la segunda parte del recorrido.	10 de septiembre de 2020
Contenido actualizado	Se han añadido patrones aws-apigateway-kinesisstreams, aws-events-rule-sns y aws-events-rule-sqs.	10 de septiembre de 2020
Contenido actualizado	Se agregó el patrón aws-sns-sqs; actualizaciones de todos los patrones SNS; correccio nes tipográficas menores.	2 de septiembre de 2020
Contenido actualizado	Se han corregido los nombres de los módulos para el patrón aws-sqs-lambda.	31 de agosto de 2020
Contenido actualizado	Se ha corregido el nombre del módulo Python para el patrón aws-dynamodb-stream-lambda-elasticsearch-kibana.	31 de agosto de 2020
Contenido actualizado	Se actualizaron los valores predeterminados para los patrones de Lambda; otras actualizaciones menores.	27 de agosto de 2020
Contenido actualizado	Propiedades públicas actualizadas para patrones de S3; valores predeterminados actualizados para patrones de DynamoDB.	10 de agosto de 2020
Contenido actualizado	Se han actualizado varios patrones para resaltar la aplicación predeterminada del cifrado en tránsito.	4 de agosto de 2020

Contenido actualizado	Se agregó el patrón aws-lambda-sqs-lambda; se mejoraron las instrucciones de configuración en la guía de introducción; se actualizaron todos los patrones para que los recursos adicionales estén disponibles a través de propiedades públicas.	27 de julio de 2020
Contenido actualizado	Se agregó el patrón aws-lambda-sqs; otras actualizaciones menores.	20 de julio de 2020
Contenido actualizado	Se eliminaron las propiedades DeployLambDA y DeployBucket de los patrones relevantes; otras actualizaciones menores.	9 de julio de 2020
Contenido actualizado	Se agregó el patrón aws-lambda-step-function y se corrigieron errores tipográficos menores.	7 de julio de 2020
Contenido actualizado	¿Se ha añadido a ExistingTableObj? para seleccionar patrones de DynamoDB.	25 de junio de 2020
Contenido actualizado	Varias correcciones de texto y correcciones para vínculos rotos.	23 de junio de 2020
Versión inicial	AWS Solutions Constructs se ponen a disposición del público.	22 de junio de 2020

Notices

Los clientes son responsables de hacer su propia evaluación independiente de la información contenida en este documento. Este documento: (a) es únicamente con fines informativos, (b) representa las ofertas y prácticas actuales de productos de AWS, que están sujetas a cambios sin previo aviso, y (c) no crea ningún compromiso o garantía de AWS y sus filiales, proveedores o licenciantes. Los productos o servicios de AWS se proporcionan «tal cual» sin garantías, representaciones ni condiciones de ningún tipo, ya sean explícitas o implícitas. Las responsabilidades y obligaciones de AWS con respecto a sus clientes se controlan mediante los acuerdos de AWS y este documento no forma parte ni modifica ningún acuerdo entre AWS y sus clientes.

© 2020 Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.