



Soluzioni AWS

Costrutti delle soluzioni AWS



Costrutti delle soluzioni AWS: Soluzioni AWS

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

I marchi e il trade dress di Amazon non possono essere utilizzati in relazione ad alcun prodotto o servizio che non sia di Amazon, in alcun modo che possa causare confusione tra i clienti, né in alcun modo che possa denigrare o screditare Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

| | |
|--|----|
| Panoramica | 1 |
| Che cos'è AWS Solutions Costrutti? | 1 |
| Perché utilizzare AWS Solutions Constructs? | 1 |
| Nozioni di base | 3 |
| Prerequisiti | 3 |
| Installazione di AWS CDK | 4 |
| Utilizzo di AWS Solutions Constructs | 4 |
| Procedura dettagliata - Parte 1 | 5 |
| Costrutti di Hello | 5 |
| Creazione della directory delle app e inizializzazione del CDK AWS | 6 |
| Aggiorna le dipendenze di base di | 7 |
| Codice del gestore Lambda | 9 |
| Installare le dipendenze di AWS CDK e AWS Solutions Constructs | 10 |
| Aggiungi un gateway API Amazon/modello AWS Lambda al tuo stack | 12 |
| Distribuzione di cdk | 18 |
| Output dello stack | 19 |
| Esecuzione del test dell'app | 19 |
| Procedura dettagliata - Parte 2 | 19 |
| Codice Lambda del contatore di colpo | 20 |
| Installare le nuove dipendenze | 22 |
| Definisci le risorse | 23 |
| Review Changes | 36 |
| Distribuzione cdk | 37 |
| Output dello stack | 38 |
| Esecuzione del test dell'app | 38 |
| Casi d'uso: | 39 |
| Sito Web AWS statico S3 | 40 |
| Gestore di immagini senza server semplice AWS | 40 |
| App Web senza server AWS | 40 |
| Riferimento alle API | 42 |
| Modules | 42 |
| Contenuto del modulo | 42 |
| aws-apigateway-dynamodb | 43 |
| Panoramica | 43 |

| | |
|--|----|
| Inizializzazione | 44 |
| Puntelli di costruzione modelli | 44 |
| Proprietà modelli | 46 |
| Impostazioni predefinite | 47 |
| Architettura | 48 |
| GitHub | 48 |
| aws-apigateway-iot | 48 |
| Panoramica | 49 |
| Inizializzatore | 50 |
| Forme di costruzione modelli | 50 |
| Proprietà modello | 51 |
| Impostazioni predefinite | 52 |
| Architettura | 55 |
| Esempi | 55 |
| GitHub | 57 |
| aws-apigateway-kinesisstream | 57 |
| Panoramica | 58 |
| Inizializzazione | 58 |
| Puntelli di costruzione modelli | 59 |
| Proprietà modelli | 60 |
| Uso dell'API di esempio | 61 |
| Impostazioni predefinite | 62 |
| Architettura | 63 |
| GitHub | 63 |
| aws-apigateway-lambda | 63 |
| Panoramica | 64 |
| Inizializzazione | 65 |
| Puntelli di costruzione modelli | 65 |
| Proprietà modelli | 66 |
| Impostazioni predefinite | 66 |
| Architettura | 67 |
| GitHub | 68 |
| aws-apigateway-sagemakerendpoint | 68 |
| Panoramica | 69 |
| Inizializzazione | 69 |
| Puntelli di costruzione modelli | 70 |

| | |
|--|----|
| Proprietà modelli | 71 |
| Uso di API di esempio | 61 |
| Impostazioni predefinite | 72 |
| Architettura | 73 |
| GitHub | 73 |
| aws-apigateway-sqs | 73 |
| Panoramica | 74 |
| Inizializzazione | 74 |
| Puntelli di costruzione modelli | 75 |
| Proprietà modelli | 77 |
| Uso di API di esempio | 61 |
| Impostazioni predefinite | 78 |
| Architettura | 79 |
| GitHub | 79 |
| aws-cloudfront-apigateway | 79 |
| Panoramica | 80 |
| Inizializzazione | 81 |
| Puntelli di costruzione modelli | 81 |
| Proprietà modelli | 82 |
| Impostazioni predefinite | 83 |
| Architettura | 83 |
| GitHub | 84 |
| aws-cloudfront-apigateway-lambda | 84 |
| Panoramica | 85 |
| Inizializzazione | 85 |
| Puntelli di costruzione modelli | 85 |
| Proprietà modelli | 87 |
| Impostazioni predefinite | 88 |
| Architettura | 89 |
| GitHub | 89 |
| aws-cloudfront-mediastore | 89 |
| Panoramica | 90 |
| Inizializzazione | 90 |
| Puntelli di costruzione modelli | 91 |
| Proprietà modelli | 92 |
| Impostazioni predefinite | 92 |

| | |
|---|-----|
| Architettura | 94 |
| GitHub | 94 |
| aws-cloudfront-s3 | 94 |
| Panoramica | 95 |
| Inizializzazione | 95 |
| Puntelli di costruzione modelli | 96 |
| Proprietà modelli | 97 |
| Impostazioni predefinite | 97 |
| Architettura | 98 |
| GitHub | 99 |
| aws-cognito-apigateway-lambda | 99 |
| Panoramica | 80 |
| Inizializzazione | 101 |
| Puntelli di costruzione modelli | 101 |
| Proprietà modelli | 102 |
| Impostazioni predefinite | 103 |
| Architettura | 105 |
| GitHub | 105 |
| AWS-Dynamodb-stream-lambda | 106 |
| Panoramica | 106 |
| Inizializzatore | 107 |
| Puntelli di costruzione modelli | 107 |
| Proprietà modelli | 108 |
| valida e completa | 108 |
| Impostazioni predefinite | 109 |
| Architettura | 110 |
| GitHub | 110 |
| aws-dynamodb-stream-lambda-elasticsearch-kibana | 110 |
| Panoramica | 111 |
| Inizializzatore | 112 |
| Puntelli di costruzione modelli | 112 |
| Proprietà pattern | 113 |
| valida e completa | 114 |
| Impostazioni predefinite | 114 |
| Architettura | 116 |
| GitHub | 116 |

| | |
|--|-----|
| aws-eventi-rule-kinesisfirehose-s3 | 116 |
| Panoramica | 117 |
| Inizializzatore | 118 |
| Puntelli di costruzione modelli | 118 |
| Proprietà pattern | 119 |
| Impostazioni predefinite | 120 |
| Architettura | 121 |
| GitHub | 121 |
| aws-eventi-rule-kinesisstream | 121 |
| Panoramica | 122 |
| Inizializzazione | 123 |
| Parametri Costrutto | 123 |
| Proprietà modelli | 124 |
| Impostazioni predefinite | 124 |
| Architettura | 125 |
| GitHub | 125 |
| aws-eventi-rule-lambda | 125 |
| Panoramica | 126 |
| Inizializzazione | 127 |
| Puntelli di costruzione modelli | 127 |
| Proprietà modelli | 128 |
| Impostazioni predefinite | 128 |
| Architettura | 129 |
| GitHub | 129 |
| aws-eventi-rule-sns | 129 |
| Panoramica | 130 |
| Inizializzazione | 131 |
| Puntelli di costruzione modelli | 131 |
| Proprietà modelli | 132 |
| Impostazioni predefinite | 133 |
| Architettura | 134 |
| GitHub | 134 |
| aws-eventi-rule-sqs | 134 |
| Panoramica | 135 |
| Inizializzazione | 136 |
| Puntelli di costruzione modelli | 136 |

| | |
|---------------------------------------|-----|
| Proprietà modelli | 138 |
| Impostazioni predefinite | 139 |
| Architettura | 140 |
| GitHub | 140 |
| aws-events-rule-step-function | 140 |
| Panoramica | 141 |
| Inizializzatore | 142 |
| Puntelli di costruzione modelli | 142 |
| Proprietà modelli | 143 |
| Impostazioni predefinite | 143 |
| Architettura | 144 |
| GitHub | 144 |
| aws-iot-kinesisfirehose-s3 | 144 |
| Panoramica | 145 |
| Inizializzatore | 146 |
| Puntelli di costruzione modelli | 146 |
| Proprietà modelli | 147 |
| Impostazioni predefinite | 148 |
| Architettura | 149 |
| GitHub | 149 |
| aws-iot-lambda | 150 |
| Panoramica | 150 |
| Inizializzatore | 151 |
| Puntelli di costruzione modelli | 151 |
| Proprietà modelli | 152 |
| Impostazioni predefinite | 152 |
| Architettura | 153 |
| GitHub | 153 |
| aws-iot-lambda-dynamodb | 154 |
| Panoramica | 154 |
| Inizializzatore | 155 |
| Puntelli di costruzione modelli | 155 |
| Proprietà modelli | 156 |
| Impostazioni predefinite | 157 |
| Architettura | 158 |
| GitHub | 158 |

| | |
|---|-----|
| aws-kinesisfirehose-s3 | 158 |
| Panoramica | 159 |
| Inizializzatore | 159 |
| Puntelli di costruzione modelli | 160 |
| Proprietà pattern | 161 |
| Impostazioni predefinite | 161 |
| Architettura | 162 |
| GitHub | 163 |
| aws-kinesisfirehose-s3-e-kinesisanalytics | 163 |
| Panoramica | 164 |
| Inizializzazione | 165 |
| Puntelli di costruzione modelli | 165 |
| Proprietà modelli | 166 |
| Impostazioni predefinite | 167 |
| Architettura | 168 |
| GitHub | 168 |
| aws-kinesisstreams-colla | 168 |
| Panoramica | 169 |
| Inizializzazione | 170 |
| Puntelli di costruzione del modello | 171 |
| SinkDataStoreProps | 172 |
| SinkStoreType | 173 |
| Impostazioni predefinite | 173 |
| Architettura | 175 |
| GitHub | 175 |
| aws-kinesisstreams-kinesisfirehose-s3 | 175 |
| Panoramica | 176 |
| Inizializzazione | 176 |
| Schema di costruzione | 177 |
| Proprietà modelli | 178 |
| Impostazioni predefinite | 179 |
| Architettura | 180 |
| GitHub | 180 |
| aws-kinesisstreams-lambda | 181 |
| Panoramica | 181 |
| Inizializzatore | 182 |

| | |
|--|-----|
| Parametri di costruzione modelli | 182 |
| Proprietà modelli | 183 |
| Impostazioni predefinite | 184 |
| Architettura | 185 |
| GitHub | 185 |
| aws-lambda-dynamodb | 185 |
| Panoramica | 186 |
| Inizializzatore | 187 |
| Puntelli di costruzione modelli | 187 |
| Proprietà modelli | 190 |
| Impostazioni predefinite | 190 |
| Architettura | 191 |
| GitHub | 191 |
| aws-lambda-elasticsearch-kibana | 192 |
| Panoramica | 192 |
| Inizializzazione | 193 |
| Puntelli di costruzione modelli | 193 |
| Proprietà modelli | 195 |
| valida e completa | 195 |
| Impostazioni predefinite | 196 |
| Architettura | 197 |
| GitHub | 197 |
| aws-lambda-s3 | 198 |
| Panoramica | 198 |
| Inizializzazione | 199 |
| Puntelli di costruzione modelli | 199 |
| Proprietà dei modelli | 203 |
| Impostazioni predefinite | 204 |
| Architettura | 205 |
| GitHub | 205 |
| aws-lambda-ssmstringparameter | 205 |
| Panoramica | 206 |
| Inizializzatore | 207 |
| Puntelli di costruzione modelli | 207 |
| Proprietà dei modelli | 211 |
| Impostazioni predefinite | 211 |

| | |
|---------------------------------------|-----|
| Architettura | 212 |
| GitHub | 212 |
| aws-lambda-sagemakerendpoint | 212 |
| Panoramica | 213 |
| Inizializzatore | 214 |
| Puntelli di costruzione modelli | 214 |
| Proprietà modelli | 218 |
| Impostazioni predefinite | 218 |
| Architettura | 220 |
| GitHub | 220 |
| aws-lambda-secretsmanager | 220 |
| Panoramica | 221 |
| Inizializzazione | 222 |
| Puntelli di costruzione modelli | 222 |
| Proprietà modelli | 225 |
| Impostazioni predefinite | 226 |
| Architettura | 227 |
| GitHub | 227 |
| aws-lambda-sns | 227 |
| Panoramica | 228 |
| Inizializzazione | 229 |
| Puntelli di costruzione modelli | 229 |
| Proprietà modelli | 232 |
| Impostazioni predefinite | 233 |
| Architettura | 234 |
| GitHub | 234 |
| aws-lambda-sqs | 234 |
| Panoramica | 235 |
| Inizializzazione | 235 |
| Puntelli di costruzione modelli | 236 |
| Proprietà del modello | 240 |
| Impostazioni predefinite | 241 |
| Architettura | 242 |
| GitHub | 242 |
| aws-lambda-sqs-lambda | 243 |
| Panoramica | 243 |

| | |
|---------------------------------------|-----|
| Inizializzatore | 244 |
| Puntelli di costruzione modelli | 244 |
| Proprietà modelli | 247 |
| Impostazioni predefinite | 247 |
| Architettura | 248 |
| GitHub | 249 |
| aws-lambda-step-funzione | 249 |
| Panoramica | 250 |
| Inizializzazione | 250 |
| Schema di costruzione | 251 |
| Proprietà modelli | 252 |
| Impostazioni predefinite | 252 |
| Architettura | 253 |
| GitHub | 253 |
| aws-s3-lambda | 253 |
| Panoramica | 254 |
| Inizializzazione | 255 |
| Puntelli di costruzione modelli | 255 |
| Proprietà dei modelli | 256 |
| Impostazioni predefinite | 256 |
| Architettura | 258 |
| GitHub | 258 |
| aws-s3-sqs | 258 |
| Panoramica | 259 |
| Inizializzazione | 259 |
| Puntelli di costruzione modelli | 260 |
| Proprietà modelli | 262 |
| Impostazioni predefinite | 263 |
| Architettura | 264 |
| GitHub | 264 |
| aws-s3-step-funzione | 264 |
| Panoramica | 265 |
| Inizializzazione | 266 |
| Puntelli di costruzione modelli | 266 |
| Proprietà modelli | 267 |
| Impostazioni predefinite | 268 |

| | |
|---|--------|
| Architettura | 270 |
| GitHub | 270 |
| aws-sns-lambda | 270 |
| Panoramica | 271 |
| Inizializzazione | 272 |
| Puntelli di costruzione modelli | 272 |
| Proprietà modelli | 273 |
| Impostazioni predefinite | 273 |
| Architettura | 274 |
| GitHub | 274 |
| aws-sns-sqs | 274 |
| Panoramica | 275 |
| Inizializzazione | 276 |
| Puntelli di costruzione modelli | 276 |
| Proprietà modelli | 278 |
| Impostazioni predefinite | 278 |
| Architettura | 279 |
| GitHub | 280 |
| aws-sqs-lambda | 280 |
| Panoramica | 281 |
| Inizializzazione | 281 |
| Puntelli di costruzione modelli | 281 |
| Proprietà modelli | 283 |
| Impostazioni predefinite | 284 |
| Architettura | 285 |
| GitHub | 285 |
| core | 285 |
| Proprietà predefinite per i costrutti CDK AWS | 286 |
| Sostituisci le proprietà predefinite | 286 |
| Avvisi della sostituzione della proprietà | 287 |
| Revisioni del documento | 288 |
| Note | 293 |
| | ccxciv |

Costrutti di soluzioni AWS

Publication Date: Maggio 2021([Revisioni del documento](#))

Che cos'è AWS Solutions Costrutti?

AWS Solutions Constructs (Constructs) è un'estensione open-source del [AWS Cloud Development Kit \(AWS CDK\)](#) che fornisce modelli multi-servizio e ben progettati per definire rapidamente le soluzioni nel codice per creare infrastrutture prevedibili e ripetibili. L'obiettivo è quello di accelerare l'esperienza per gli sviluppatori nella creazione di soluzioni di qualsiasi dimensione utilizzando definizioni basate su modelli per la loro architettura.

Usa i costrutti di soluzioni AWS per definire le tue soluzioni in un linguaggio di programmazione familiare. Le soluzioni AWS costrutti supporta TypeScript, JavaScript, Python e Java in questo momento.

Per sfogliare il catalogo completo dei modelli AWS Solutions Constructs, [Clicca QUI](#): .

Perché utilizzare AWS Solutions Constructs?

Con la velocità di innovazione dei provider cloud, conoscere e comprendere le best practice e garantire che siano implementate correttamente in tutta la soluzione può essere scoraggiante. Constructs consente di combinare modelli predefiniti e ben progettati e casi d'uso che eseguono azioni comuni utilizzando i servizi cloud in modo scalabile e sicuro. Poiché Constructs fornisce una libreria per i moderni linguaggi di programmazione, è possibile applicare competenze di sviluppo esistenti e strumenti familiari al compito di costruire un'infrastruttura cloud ben progettata per le proprie soluzioni.

Altri vantaggi di AWS Solutions Constructs includono:

- È basato sul framework di Cloud Development Kit software open source AWS (AWS CDK).
- Utilizzare la logica (se istruzioni, for-loop, ecc.) Quando si definisce l'infrastruttura della soluzione.
- Utilizzare tecniche orientate agli oggetti per creare un modello del sistema.
- Definisci astrazioni di alto livello, condividile e pubblicale nel tuo team, azienda o community.
- Organizza le tue soluzioni in moduli logici.
- Condividi e riutilizza la tua soluzione come libreria.
- Testare il codice dell'infrastruttura utilizzando protocolli standard di settore.

- Utilizzare il flusso di lavoro di revisione del codice esistente.

Lo scopo di AWS Solutions Constructs è ridurre la complessità e la logica di colla richiesta quando si integrano modelli comuni ben progettati per raggiungere gli obiettivi della soluzione su AWS.

Utilizzo di AWS Solutions Constructs

In questo argomento viene descritto come installare e configurare AWS Cloud Development Kit (AWS CDK), AWS Solutions Constructs e creare la prima app AWS CDK utilizzando i modelli AWS Solutions Constructs.

Note

AWS Solutions Constructs è supportato su AWS CDK versioni $\geq 1.46.0$.

Tip

Vuoi scavare più a fondo? Prova la [Workshop CDK](#) per un tour più approfondito di un progetto del mondo reale.

Tip




Per ulteriori informazioni su come usare l'AWS Cloud Development Kit (AWS CDK), consulta la [Guida per sviluppatori di AWS CDK](#): .

Prerequisites

AWS Solutions Constructs è basato su AWS CDK, quindi è necessario installare Node.js ($> = 10.3.0$), anche quelli che lavorano in lingue diverse da TypeScript o JavaScript. Questo perché [ilAWS CDK](#)e AWS Solutions Constructs sono sviluppati in TypeScript ed eseguiti su Node.js. I binding per altre lingue supportate utilizzano questo back-end e toolset.

È necessario fornire le credenziali e un'area AWS per utilizzare l'interfaccia della riga di comando CDK di AWS, come descritto in [Specifiche delle credenziali e dell'area geografica](#).

Altri prerequisiti dipendono dal linguaggio di sviluppo, come indicato di seguito.

| Linguaggio | Prerequisiti |
|--|------------------------|
|  | Python >= 3.6 P |
|  t | TypeScript >= 2.7 T |
|  | Java >= 1.8 J; |

Installazione di AWS CDK

Per installare e configurare AWS CDK, fare riferimento alla Guida per sviluppatori CDK AWS

[-Installazione di AWS CDK:](#) .

Utilizzo di AWS Solutions Constructs

Il flusso di lavoro tipico per la creazione di una nuova app quando si lavora con AWS Solutions Constructs segue lo stesso approccio di AWS CDK.

1. Creare la directory dell'app.
2. Inizializzare l'app.
3. Aggiungere le dipendenze del modello AWS Solutions Constructs.
4. Aggiungere codice aggiuntivo all'app.
5. Compilare l'app, se necessario.
6. Distribuire le risorse definite nell'app.
7. Eseguire il test dell'app.

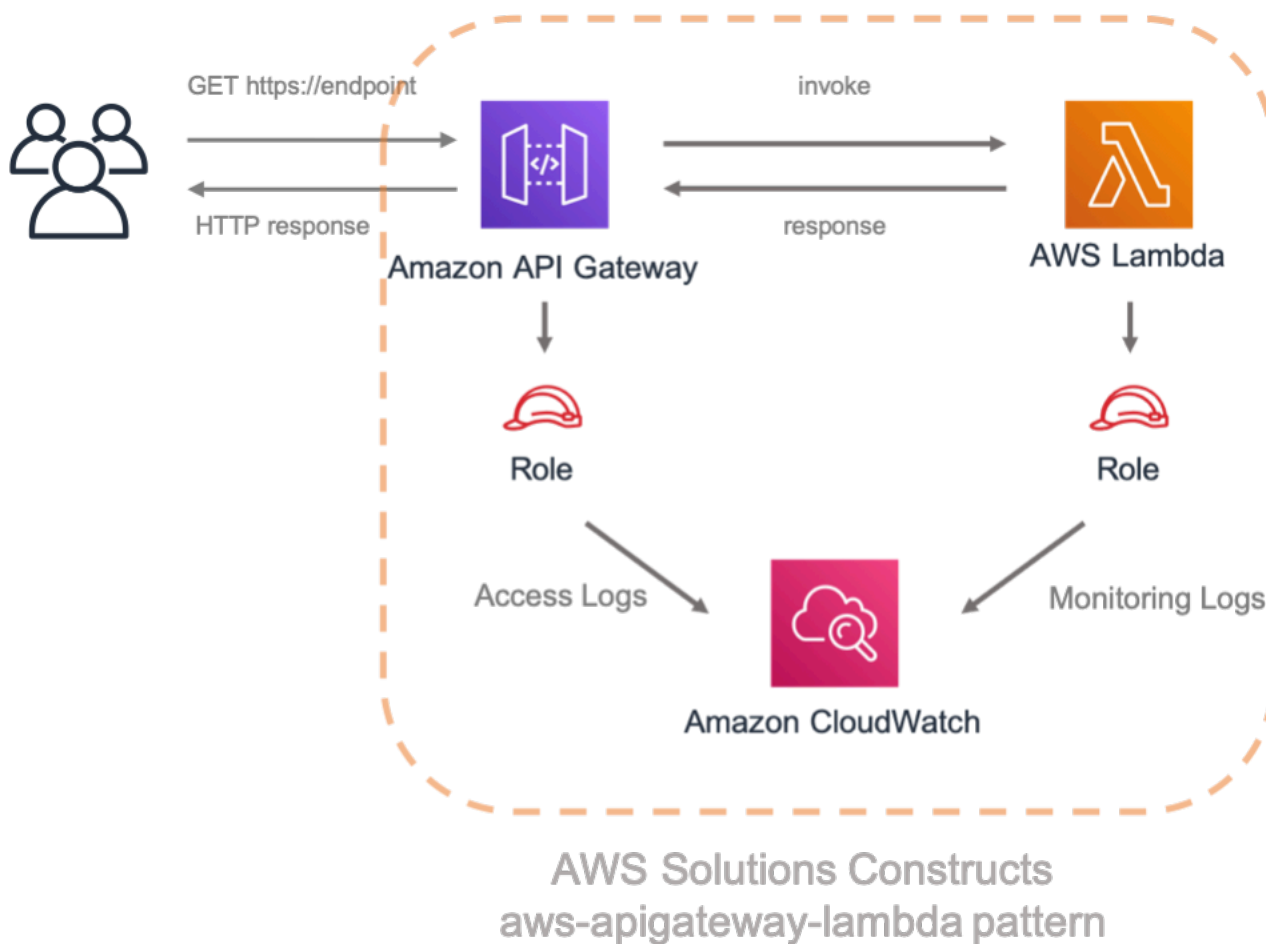
Se ci sono problemi, eseguire il ciclo attraverso la modifica, compilare (se necessario), distribuire e testare nuovamente.

Procedura dettagliata - Parte 1

Note

Costrutti di AWS Solutions è supportato sulle versioni CDK $\geq 1.46.0$.

Questo tutorial illustra come creare e distribuire una semplice app AWS CDK «Hello Constructs» che utilizza un modello di AWS Solutions Constructs, dall'inizializzazione del progetto alla distribuzione del modello AWS CloudFormation risultante. L'app Hello Constructs creerà la seguente semplice soluzione:



Costrutti di Hello

Iniziamo a costruire la nostra prima app CDK AWS utilizzando lo sviluppo basato su pattern.

Note

Questa è una modifica di esempio di Hello CDK! dagli [Workshop CDK](#): . Se questa è la prima volta che utilizzi AWS CDK, ti consigliamo di iniziare con questo workshop per una procedura dettagliata pratica e come sfruttare il CDK nella costruzione di un progetto reale.

Creazione della directory delle app e inizializzazione del CDK AWS

Creare una directory per l'app CDK e quindi creare un'app AWS CDK in tale directory.

TypeScript

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

Python

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

Tip

Ora è il momento giusto per aprire il progetto nel tuo IDE preferito ed esplorare. Per ulteriori informazioni sulla struttura del progetto, seleziona il collegamento appropriato:

- [TypeScript](#)
- [Python](#)

Aggiorna le dipendenze di base di

Warning

Per garantire la corretta funzionalità, AWS Solutions Constructs e AWS CDK pacchetti devono utilizzare lo stesso numero di versione all'interno del progetto. Ad esempio, se si utilizza AWS Solutions Constructs v.1.52.0, è necessario utilizzare anche AWS CDK v.1.52.0.

Tip

Prendere nota della versione più recente di AWS Solutions Constructs e applicare tale numero di versione al `VERSION_NUMBER` nella procedura riportata di seguito (sia per AWS Solutions Constructs che per i pacchetti AWS CDK). Per controllare tutte le versioni pubbliche della libreria Costrutti, [Fare clic qui](#): .

TypeScript

Modificare il file `package.json` con le seguenti informazioni:

```
"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

Modificare il file `setup.py` con le seguenti informazioni:

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

Installare le dipendenze di base dei progetti.

TypeScript

```
npm install
```

Python

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

Crea ed esegui l'app e conferma che crea uno stack vuoto.

TypeScript

```
npm run build  
cdk synth
```

Python

```
cdk synth
```

Dovresti vedere uno stack come il seguente, dove `CDK-VERSION` è la versione del CDK. (Il tuo output potrebbe differire leggermente da quello mostrato qui.)

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
```

Codice del gestore Lambda

Inizieremo con il codice gestore AWS Lambda.

Creazione di una directory `lambda` nella directory root dell'albero del progetto.

TypeScript

Aggiungere un file denominato `lambda/hello.js` con i seguenti contenuti:

```
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

Aggiungere un file denominato `lambda/hello.py` con i seguenti contenuti:

```
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 {}'.format(event['path'])
    }
```

Questa è una semplice funzione Lambda che restituisce il testo «Ciao, costrutti! Hai colpito [url path]». L'output della funzione include anche il codice di stato HTTP e le intestazioni HTTP. Questi vengono utilizzati da API Gateway per formulare la risposta HTTP all'utente.

Questa Lambda è fornita in JavaScript. Per ulteriori informazioni sulla scrittura delle funzioni Lambda nella lingua di scelta, fare riferimento alla [Documentazione AWS Lambda](#): .

Installare le dipendenze di AWS CDK e AWS Solutions Constructs

AWS Solutions Constructs viene fornito con una vasta libreria di costrutti. La libreria è divisa in moduli, uno per ogni modello ben progettato. Ad esempio, se si desidera definire un'Amazon API Gateway Rest API per una funzione AWS Lambda, sarà necessario utilizzare il `aws-apigateway-lambda` Libreria di pattern.

Abbiamo anche bisogno di aggiungere la libreria di costrutto AWS Lambda e Amazon API Gateway dal CDK AWS.

Installa il modulo AWS Lambda e tutte le sue dipendenze nel nostro progetto:

Note

Ricordarsi di sostituire la versione corretta e corrispondente da utilizzare sia per AWS Solutions Constructs che per AWS CDK nei `VERSION_NUMBER` campi segnati per ogni comando. La mancata corrispondenza delle versioni tra i pacchetti può causare errori.

TypeScript

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

Python

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

Successivamente, installa il modulo Amazon API Gateway e tutte le sue dipendenze nel nostro progetto:

TypeScript

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

Python

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

Infine, installa i costrutti di soluzioni AWS `saws-apigateway-lambda` e tutte le sue dipendenze nel nostro progetto:

TypeScript

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

Python

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

Aggiungi un gateway API Amazon/modello AWS Lambda al tuo stack

Ora, definiamo il modello AWS Solutions Constructs per l'implementazione di un Amazon API Gateway con un proxy AWS Lambda.

TypeScript

Modificare il file `lib/hello-constructs.ts` con gli elementi seguenti:

```
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

Modificare il file `hello_constructs/hello_constructs_stack.py` con gli elementi seguenti:

```

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
                )
            )
        )

```

```
)
)
)
```

Questo è tutto. Questo è tutto ciò che devi fare per definire un API Gateway che proxy tutte le richieste a una funzione AWS Lambda. Confrontiamo il nostro nuovo stack con quello originale:

TypeScript

```
npm run build
cdk diff
```

Python

```
cdk diff
```

L'output dovrebbe essere simile al seguente:

```
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
  RestApiDeployment180EC503d2c6df3c8dc8b7193b98c1a0bfff4e677
[+] 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"},"/"]}}
```

Che bello. Questo semplice esempio con un modello ben progettato da AWS Solutions Constructs ha aggiunto 21 nuove risorse al tuo stack.

Distribuzione di cdk

Tip

Prima di poter distribuire la prima app AWS CDK contenente una funzione Lambda, è necessario eseguire il bootstrap dell'ambiente AWS. Questo crea un bucket di gestione temporanea che AWS CDK utilizza per distribuire stack contenenti risorse. Se questa è la prima volta che si utilizza AWS CDK per distribuire le risorse, sarà necessario eseguire il comando `cdk bootstrap` per distribuire lo stack di toolkit CDK nel tuo ambiente AWS.

Pronti per la distribuzione?

```
cdk deploy
```

Output dello stack

Al termine della distribuzione, noterai questa riga:

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

Si tratta di un output dello stack che viene aggiunto automaticamente dal modello AWS Solutions Constructs e include l'URL dell'endpoint del API Gateway.

Esecuzione del test dell'app

Proviamo a colpire questo endpoint con `curl`: . Copia l'URL ed esegui (il tuo prefisso e la tua regione saranno probabilmente diversi).

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

L'output dovrebbe essere simile al seguente:

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

Se questo è l'output che hai ricevuto, la tua app funziona!

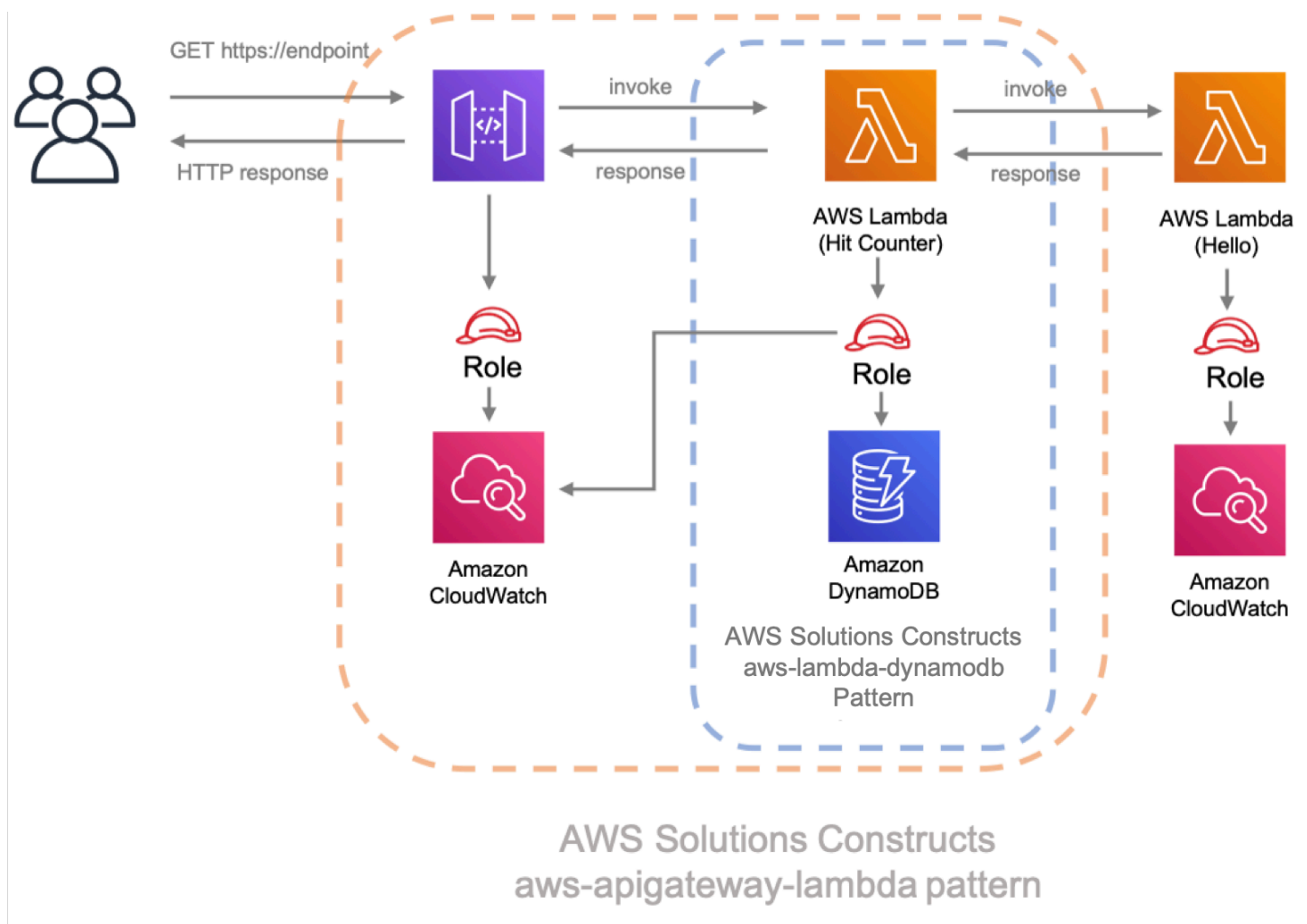
Procedura dettagliata - Parte 2

Note

AWS Solutions Constructs è supportato sulle versioni CDK \geq 1.46.0.

Questo tutorial illustra come modificare l'app «Hello Constructs» creata in [PART 1](#): . La nostra modifica aggiungerà un contatore di visite del sito utilizzando il modello AWS Lambda a DynamoDB

da AWS Solutions Constructs. La modifica dell'app Hello Constructs si tradurrà nella seguente soluzione:



Codice Lambda del contatore di colpo

Iniziamo scrivendo il codice per la funzione Hit Counter AWS Lambda. Questa funzione:

- incrementare un contatore correlato al percorso API in una tabella Amazon DynamoDB,
- richiamare la funzione Hello AWS Lambda a valle,
- e restituire la risposta all'utente finale.

TypeScript

Aggiungere un file denominato `lambda/hitcounter.js` con i seguenti contenuti:

```
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

Aggiungere un file denominato `lambda/hitcounter.py` con i seguenti contenuti:

```
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)
```

Installare le nuove dipendenze

Note

Ricordarsi di sostituire la versione corretta e corrispondente da utilizzare sia per AWS Solutions Constructs che per AWS CDK nel `VERSION_NUMBER` campi segnato per ogni comando. Questo dovrebbe essere identico al numero di versione utilizzato per le dipendenze nella prima parte di questa procedura dettagliata. La mancata corrispondenza delle versioni tra i pacchetti può causare errori.

Come al solito, dobbiamo prima installare le dipendenze di cui abbiamo bisogno per l'aggiornamento della nostra soluzione. Innanzitutto, dobbiamo installare la libreria di costruito DynamoDB:

TypeScript

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

Python

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

Infine, installa i costrutti di soluzioni AWS `aws-lambda-dynamodb` e tutte le sue dipendenze nel nostro progetto:

TypeScript

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

Python

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

Definisci le risorse

Ora, aggiorniamo il nostro codice di stack per accogliere la nostra nuova architettura.

Innanzitutto, importeremo le nostre nuove dipendenze e sposteremo la funzione «Ciao» al di fuori del `aws-apigateway-lambda` abbiamo creato nella parte 1.

TypeScript

Modificare il file `lib/hello-constructs.ts` con gli elementi seguenti:

```
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

Modificare il file `hello_constructs/hello_constructs_stack.py` con gli elementi seguenti:

```

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):

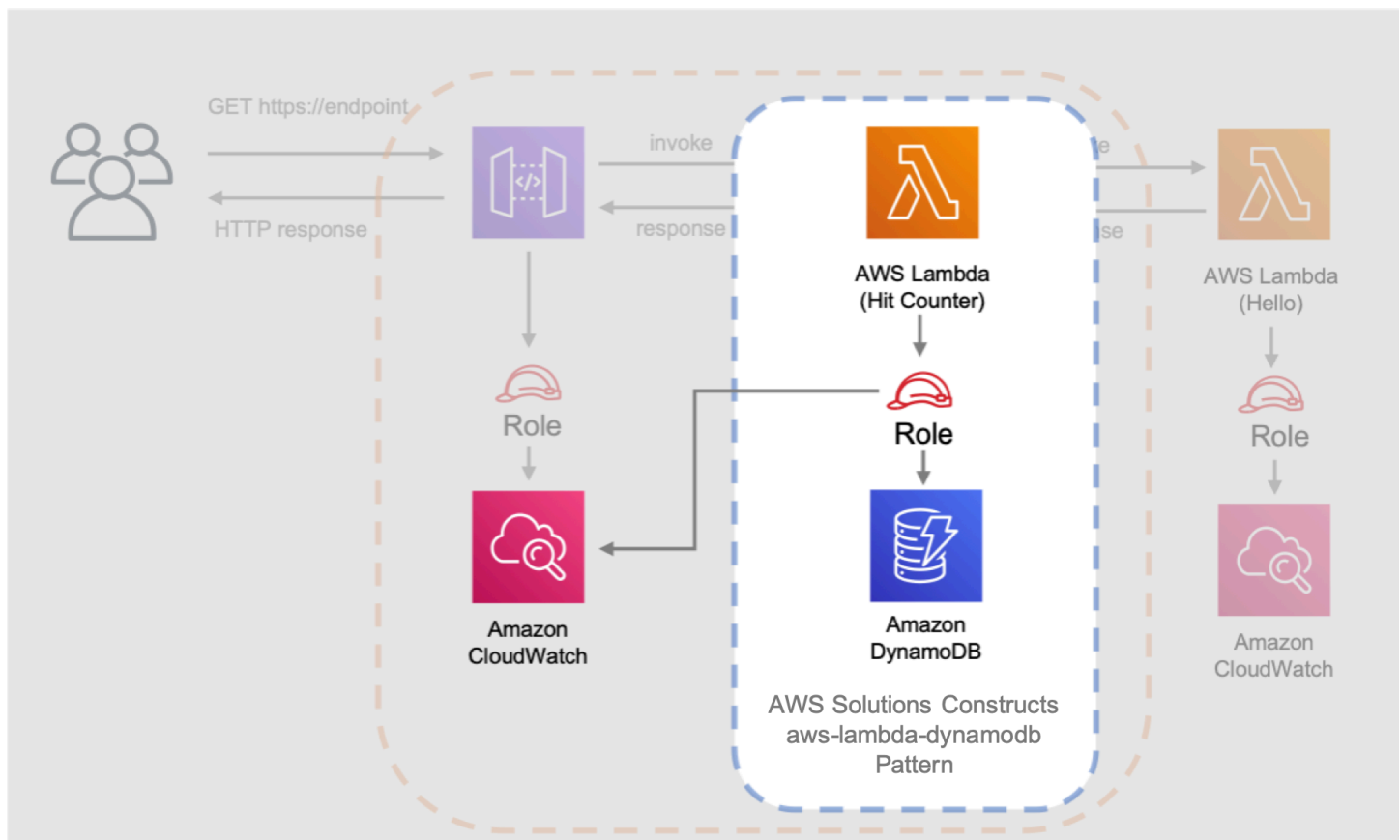
    def __init__(self, scope: core.Construct, id: str, **kwargs) -> None:
        super().__init__(scope, id, **kwargs)

        # 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
                )
            )
        )
    )
```

Successivamente, aggiungeremo il `aws-lambda-dynamodb` per costruire il servizio di contatore degli accessi per la nostra architettura aggiornata.



AWS Solutions Constructs aws-apigateway-lambda pattern

Il prossimo aggiornamento di seguito definisce le proprietà per il `aws-lambda-dynamodb` definendo la funzione AWS Lambda con il gestore Hit Counter. Inoltre, la tabella Amazon DynamoDB è definita con il nome `Hit` e una chiave di partizione di `path`.

TypeScript

Modificare il file `lib/hello-constructs.ts` con gli elementi seguenti:

```
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

Modificare il file `hello_constructs/hello_constructs_stack.py` con gli elementi seguenti:

```
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
        )
    )
)
)

```

Successivamente, dobbiamo concedere la funzione Hit Counter creata dalla `aws-lambda-dynamodb` aggiunto sopra il permesso di richiamare la nostra funzione Hello.

TypeScript

Modificare il file `lib/hello-constructs.ts` con gli elementi seguenti:

```

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

Modificare il file `hello_constructs/hello_constructs_stack.py` con gli elementi seguenti:

```

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
        )
    )
)

```

Infine, abbiamo bisogno di aggiornare il nostro `aws-apigateway-lambda` per utilizzare la nostra nuova funzione Hit Counter che è stata fornita con `aws-lambda-dynamodb` Modello di cui sopra.

TypeScript

Modificare il file `lib/hello-constructs.ts` con gli elementi seguenti:

```
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

Modificare il file `hello_constructs/hello_constructs_stack.py` con gli elementi seguenti:

```

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
        )
    )
)
```


Review Changes

Costruiamo il nostro progetto ed esaminiamo le modifiche alle nostre risorse che avverranno quando implementeremo questo:

```
npm run build
cdk diff
```

Il nostro output dovrebbe essere simile al seguente:

```
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"
[ ] ]

```

Distribuzione cdk

Ok, pronto per la distribuzione?

```
cdk deploy
```

Output dello stack

Al termine della distribuzione, noterai questa riga:

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

Esecuzione del test dell'app

Proviamo a colpire questo endpoint con arricciatura. Copia l'URL ed esegui (il tuo prefisso e la tua regione saranno probabilmente diversi).

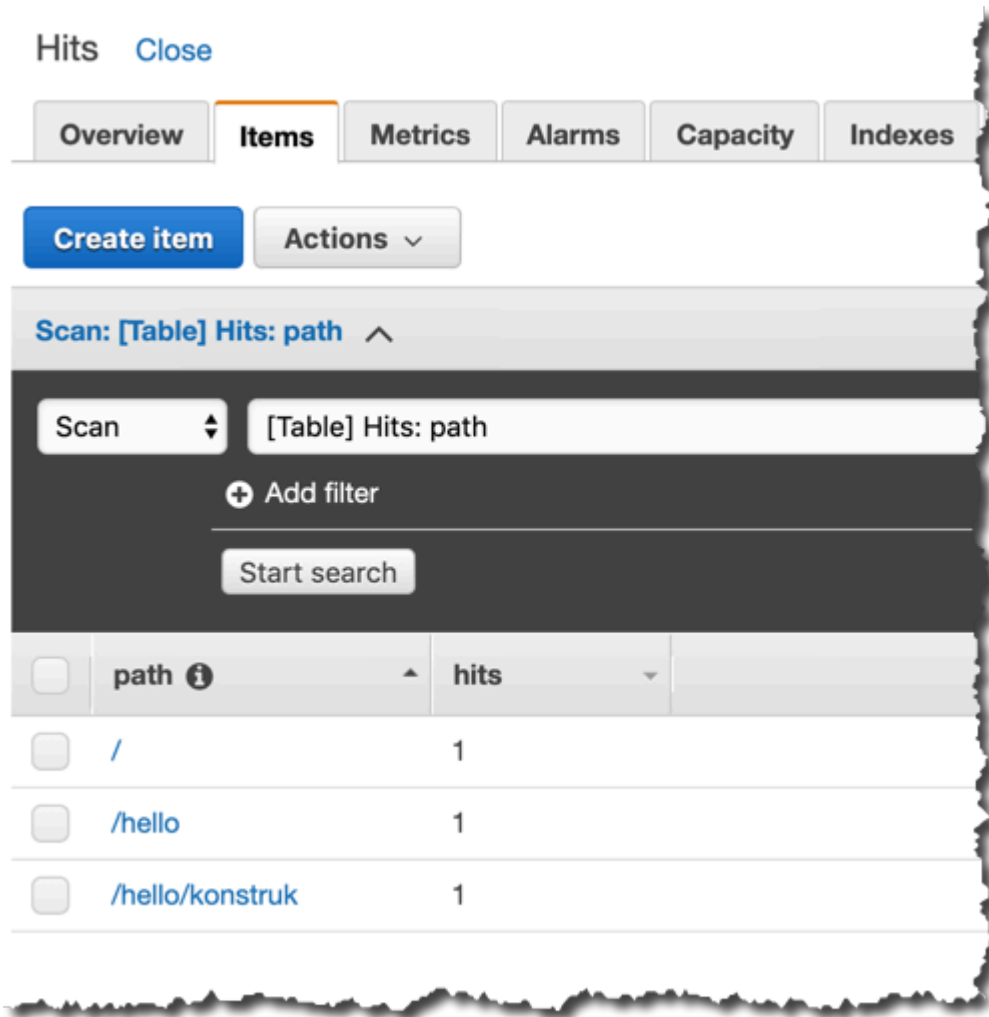
```
curl https://xxxxxxxxxx.execute-api.us-east-1.amazonaws.com/prod/
```

L'output dovrebbe essere simile al seguente:

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

Ora, rivedere gli elementiHitsTabella Amazon DynamoDB.

1. Passa alla console DynamoDB.
2. Assicurarsi di essere nella regione in cui è stata creata la tabella.
3. SelezionaTabellenel riquadro di navigazione e selezionare la casella di controlloHits (occorrenze)INTO table
4. Apri la tabella e seleziona «Elementi».
5. Dovresti vedere quanti colpi hai ottenuto per ogni percorso.



6. Prova a colpire un nuovo percorso e aggiorna la vista Elementi. Viene visualizzato un nuovo elemento con un `hits` conteggio di uno.

Se questo è l'output che hai ricevuto, la tua app funziona!

Casi d'uso:

Questa libreria include una raccolta di implementazioni di casi d'uso funzionali per dimostrare l'utilizzo dei modelli architettonici di costrutti. Questi possono essere utilizzati allo stesso modo dei modelli architettonici e possono essere concettualizzati come un'astrazione aggiuntiva di «livello superiore» di tali modelli. I seguenti casi d'uso sono forniti come esempi funzionali:

Sito Web AWS statico S3

Questo modello di caso d'uso (`aws-s3-static-website`) implementa una distribuzione Amazon CloudFront, Amazon S3 bucket e una risorsa personalizzata basata su AWS Lambda per copiare il contenuto del sito Web statico per il sito web demo Wild Rydes (parte del `aws-serverless-web-app` implementazione di)

 Codice sorgente (`aws-s3-static-website`)

https://github.com/awslabs/aws-solutions-constructs/tree/master/source/use_cases/aws-s3-static-website

Gestore di immagini senza server semplice AWS


Questo modello di caso d'uso (`aws-serverless-image-handler`) implementa una distribuzione Amazon CloudFront, un'API REST Gateway API di Amazon, una funzione AWS Lambda e le autorizzazioni e la logica necessarie per il provisioning di un'API funzionale del gestore di immagini per la pubblicazione del contenuto dell'immagine da uno o più bucket Amazon S3 all'interno dell'account di distribuzione.

 Codice sorgente (`aws-serverless-image-handler`)

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

App Web senza server AWS

Questo modello di caso d'uso (`aws-serverless-web-app`) implementa una semplice applicazione web senza server che consente agli utenti di richiedere corse unicorno dalla flotta Wild Rydes. L'applicazione presenterà agli utenti un'interfaccia utente basata su HTML per indicare la posizione in cui vorrebbero essere prelevati e si interfaccia sul back-end con un servizio web RESTful per inviare la richiesta e inviare un unicorno vicino. L'applicazione fornirà anche servizi per gli utenti di registrarsi con il servizio e accedere prima di richiedere corse.

 Codice sorgente (aws-serverless-web-app)

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

Riferimento alle API

AWS Solutions Constructs (Constructs) è un'estensione open-source di AWS Cloud Development Kit (AWS CDK) che fornisce modelli multi-servizio e ben progettati per definire rapidamente le soluzioni nel codice per creare infrastrutture prevedibili e ripetibili. L'obiettivo di Constructs è quello di accelerare l'esperienza per gli sviluppatori di creare soluzioni di qualsiasi dimensione utilizzando definizioni basate su modelli per la loro architettura.

I modelli definiti in costrutti sono astrazioni multi-servizio di alto livello di costrutti AWS CDK che hanno configurazioni predefinite basate su best practice ben architettate. La libreria è organizzata in moduli logici utilizzando tecniche orientate agli oggetti per creare ogni modello di pattern architettonico.

Il CDK è disponibile nelle seguenti lingue:

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

Modules

AWS Solutions Constructs è organizzato in diversi moduli. Essi sono chiamati in questo modo:

- **AWS:** Pacchetto di pattern ben progettato per i servizi indicati. Questo pacchetto conterrà costrutti che contengono più moduli di servizio CDK AWS per configurare il modello dato.
- **xxx:** Pacchetti che non si avviano»- AWS«sono moduli di base costrutti che vengono utilizzati per configurare i valori predefiniti delle best practice per i servizi utilizzati all'interno della libreria dei pattern.

Contenuto del modulo

I moduli contengono i seguenti tipi:

- **Modelli-** Tutti i costrutti multi-servizi di livello superiore in questa libreria.
- **Altri tipi-** Tutte le classi, interfacce, strutture ed enumerazioni non costruttive che esistono per supportare i modelli.

I modelli prendono un insieme di proprietà (input) nel loro costruttore; l'insieme di proprietà (e quali sono richieste) può essere visto nella pagina di documentazione di un modello.

La pagina di documentazione del modello elenca anche i metodi disponibili per chiamare e le proprietà che possono essere utilizzate per recuperare informazioni sul modello dopo che è stato istanziato.



aws-apigateway-dynamodb

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Versionamento semantico](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_apigateway_dynamodb</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-apigateway-dynamodb</code> |
|  Java | <code>software.amazon.awsconstructs.services.apigatewaydynamodb</code> |

Overview

Questo Construct di soluzioni AWS implementa un'Amazon API Gateway API connessa a una tabella Amazon DynamoDB.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id `string`
- props [ApiGatewayToDynamoDBProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|------------------|-------------------------------------|---|
| DynamoTableProps | dynamodb.TableProps | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per DynamoDB Table |
| ApiGatewayProps? | api.RestApiProps | Oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il API Gateway. |

| Nome | Tipo | Descrizione |
|-----------------------|---------|---|
| AllowCreateOperation | boolean | Indica se distribuire il metodo API Gateway per l'operazione di creazione sulla tabella DynamoDB. |
| CreateRequestTemplate | string | Modello di richiesta API Gateway per il metodo Create, obbligatorio se allowCreateOperation è impostato su true |
| AllowreadOperation | boolean | Indica se distribuire il metodo API Gateway per l'operazione di lettura sulla tabella DynamoDB. |
| AllowUpdateOperation | boolean | Indica se distribuire il metodo API Gateway per l'operazione di aggiornamento sulla tabella DynamoDB. |
| UpdateRequestTemplate | string | Modello di richiesta API Gateway per il metodo Update, obbligatorio se allowUpdateOperation è impostato su true |
| AllowDeleteOperation | boolean | Indica se distribuire il metodo API Gateway per l'operazione di eliminazione sulla tabella DynamoDB. |

| Nome | Tipo | Descrizione |
|----------------|---|---|
| LogGroupProps? | <u>logs.LogGroupProps</u> | Gli oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|--------------------------|--------------------------------------|--|
| ApiGateway | <u>api.RestApi</u> | Restituisce un'istanza dell'API REST del gateway API creata dal modello. |
| APigatewayCloudWatchRole | <u>iam.Role</u> | Restituisce un'istanza del ruolo IAM creato dal pattern che abilita la registrazione degli accessi dall'API REST del gateway API a CloudWatch. |
| ApiGatewayLogGroup | <u>logs.LogGroup</u> | Restituisce un'istanza del gruppo di log creato dal modello a cui vengono inviati i log di accesso API REST del API Gateway API. |
| ApiGatewayRole | <u>iam.Role</u> | Restituisce un'istanza del ruolo IAM creato dal pattern per l'API REST del gateway API. |

| Nome | Tipo | Descrizione |
|-------------|---------------------------------------|---|
| DynamoTable | <u>dynamodb.Table</u> | Restituisce un'istanza della tabella DynamoDB creata dalla serie. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

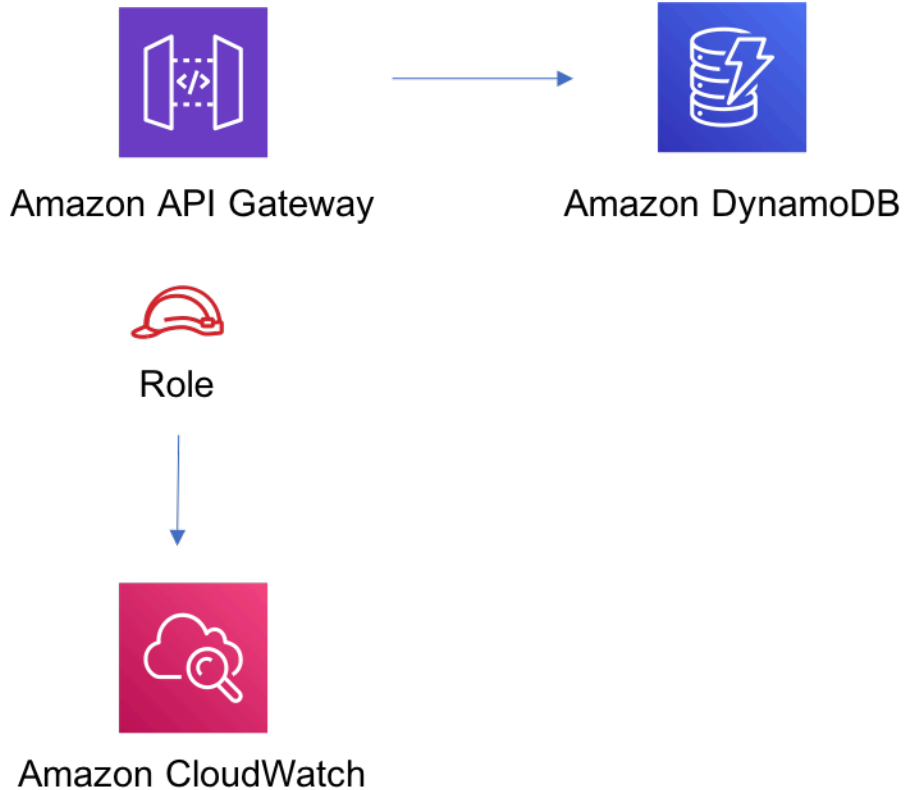
Amazon API Gateway

- Distribuzione di un endpoint API ottimizzato per gli edge
- Abilitazione della registrazione CloudWatch per API Gateway
- Configurare il ruolo IAM di accesso ai privilegi minimi per il API Gateway
- Impostare l'AuthorizationType predefinito per tutti i metodi API su IAM
- Abilita il monitoraggio X-Ray

Amazon DynamoDB

- Impostare la modalità di fatturazione per DynamoDB Table su richiesta (Paga per richiesta)
- Abilitare la crittografia lato server per la tabella DynamoDB utilizzando la chiave AWS gestita
- Crea una chiave di partizione chiamata 'id' per DynamoDB Table
- Mantieni la tabella quando si elimina lo stack CloudFormation
- Backup continui e ripristino point-in-time

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:




[@aws -solutions-buils/aws-apigateway-dynamodb](https://github.com/aws-solutions-buils/aws-apigateway-dynamodb)

aws-apigateway-iot

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Controllo semantico delle versioni](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_apigateway_iot</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-apigateway-iot</code> |
|  Java | <code>software.amazon.awsconstructs.services.apigatewayiot</code> |

Overview

Questo Construct di soluzioni AWS implementa un'API REST del gateway API Amazon connessa al modello AWS IoT.

Questo costrutto crea un proxy HTTPS scalabile tra API Gateway e AWS IoT. Ciò risulta utile quando si desidera consentire ai dispositivi legacy che non supportano il protocollo MQTT o MQTT/WebSocket di interagire con la piattaforma AWS IoT.

Questa implementazione consente di pubblicare messaggi di sola scrittura su determinati argomenti MQTT e supporta anche gli aggiornamenti shadow dei dispositivi HTTPS agli elementi consentiti nel Registro di sistema dei dispositivi. Non coinvolge le funzioni Lambda per il proxy dei messaggi, e si basa invece sull'integrazione diretta API Gateway all'AWS IoT che supporta sia i messaggi JSON che i messaggi binari.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- `scope` [Construct](#)
- `id` `string`
- `props` [ApiGatewayToIotProps](#)

Forme di costruzione modelli

| Nome | Tipo | Descrizione |
|---------------------------------------|--------------------------|---|
| <code>IOtendPoint</code> | <code>string</code> | Sottodominio dell'endpoint AWS IoT per integrare il API Gateway con (ad esempio <code>a1234567890123-ats</code>). |
| <code>ApiGatewayCreateApiKey?</code> | <code>boolean</code> | Se è impostato su <code>true</code> , viene creata una chiave API e associata a un <code>UsagePlan</code> . L'utente deve specificare l'intestazione <code>`x-api-key`</code> durante l'accesso a <code>RestApi</code> . Valore predefinito impostato su <code>false</code> . |
| <code>ApiGatewayExecutionRole?</code> | iam.Role | Il ruolo IAM utilizzato da API Gateway per accedere a AWS IoT. Se non viene specificato, viene creato un ruolo predefini |

| Nome | Tipo | Descrizione |
|------------------|------------------------------------|---|
| | | to con accesso jolly (*) a tutti gli argomenti e le cose. |
| ApiGatewayProps? | api.restApiProps | Oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per l'API REST del gateway API. |
| LogGroupProps? | logs.LogGroupProps | Argomenti facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà modello

| Nome | Tipo | Descrizione |
|--------------------------|-------------------------------|--|
| ApiGateway | api.RestApi | Restituisce un'istanza dell'API REST del gateway API creata dal modello. |
| APigatewayCloudWatchRole | iam.Role | Restituisce un'istanza del ruolo IAM creato dal pattern che abilita la registrazione degli accessi dall'API REST del gateway API a CloudWatch. |
| ApiGatewayLogGroup | logs.LogGroup | Restituisce un'istanza del gruppo di log creato dal modello a cui vengono inviati i log di accesso API REST del API Gateway API. |

| Nome | Tipo | Descrizione |
|----------------|--------------------------|---|
| ApiGatewayRole | iam.Role | Restituisce un'istanza del ruolo IAM creato dal pattern per l'API REST del gateway API. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Amazon API Gateway

- Distribuisci di un endpoint API ottimizzato per gli edge
- Crea risorse API conPOSTMetodo per pubblicare messaggi in Argomenti IoT
- Crea risorse API conPOSTMetodo per pubblicare messaggi inThingShadoweNamedShadows
- Attivazione della registrazione CloudWatch per API Gateway
- Configurare il ruolo IAM per API Gateway con accesso a tutti gli argomenti
- Impostare l'AuthorizationType predefinito per tutti i metodi API su IAM
- Attiva traccia X-Ray
- Crea un UsagePlan e associa aprod fase

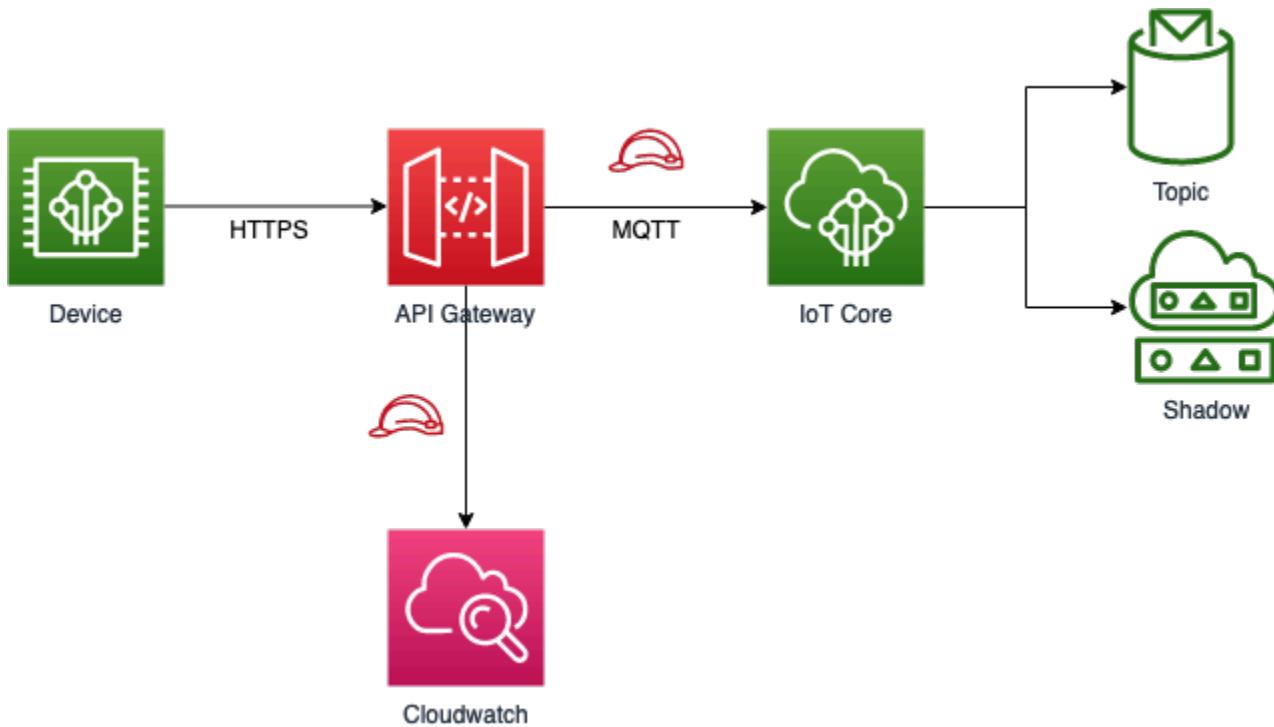
Di seguito è riportata una descrizione delle diverse risorse e metodi esposti dal API Gateway dopo la distribuzione del Costrutto. Consultare la [Esempi](#) per ulteriori informazioni su come testare facilmente questi endpoint tramite `curl`:

| Metodo | Risorsa | Parametri di query | Codici restituiti | Descrizione |
|--------|-----------------------|--------------------|-------------------|---|
| POST | /message/ <topics> | qos | 200/403/500 | Chiamando questo endpoint, è necessari o passare gli argomenti su cui si desidera |

| Metodo | Risorsa | Parametri di query | Codici restituiti | Descrizione |
|--------|---------------------|--------------------|-------------------|---|
| | | | | pubblicare (ad es. /message/device/fo o »). |
| POST | /shadow/<thingName> | Nessuna | 200/403/500 | <p>Questo percorso permette di aggiornare il documento shadow di una cosa, dato il suo thingName</p> <p>Utilizzando il tipo di copia shadow senza nome (classico). Il corpo deve essere conforme alla struttura ombreggiata standard comprende ntestatenodo e associato desiredereportedNode</p> <p>Consultare laAggiornamento di una copia shadow per un esempio.</p> |

| Metodo | Risorsa | Parametri di query | Codici restituiti | Descrizione |
|--------|----------------------------------|--------------------|-------------------|---|
| POST | /shadow/<thingName>/<shadowName> | Nessuna | 200/403/500 | <p>Questo percorso consente di aggiornare il documento shadow denominato di una cosa, dato il suo thingName e il suo shadowName e utilizzando il tipo di ombreggiatura con nome. Il corpo deve essere conforme alla struttura ombreggiatura standard comprende il nodo e associato desiderato reportedNode. Consultare la Guida all'aggiornamento di una copia shadow per un esempio.</p> |

Architecture



Examples

Gli esempi seguenti funzionano solo con `API_KEY` tipi di autenticazione, dal momento che l'autorizzazione IAM richiede anche un token Sigv4 da specificare, assicurarsi che il `apiGatewayCreateApiKey` dei tuoi oggetti di scena Construct è impostata su `true` durante la distribuzione dello stack, altrimenti gli esempi seguenti non funzioneranno.

Pubblicazione di un messaggio

È possibile utilizzare `curl` per pubblicare un messaggio su diversi argomenti MQTT utilizzando l'API HTTPS. L'esempio seguente pubblicherà un messaggio sul `device/foo` argomento.

```
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: Sostituisci il `stage-id`, `region`, e `api-key` con i valori di distribuzione.

È possibile concatenare i nomi degli argomenti nell'URL e l'API accetta fino a 7 sottoargomenti su cui è possibile pubblicare. Ad esempio, nell'esempio seguente viene pubblicato un messaggio nell'argomentodevice/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"}
```

Aggiornamento di una copia shadow

Per aggiornare il documento shadow associato a una determinata cosa, è possibile emettere una richiesta di stato shadow utilizzando un nome di cosa. Consultare l'esempio seguente su come aggiornare una copia shadow di un oggetto.

```
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" }}}'
```

Aggiornamento di una copia shadow

Per aggiornare il documento shadow associato all'ombra denominata di una determinata cosa, è possibile emettere una richiesta di stato shadow utilizzando un nome di cosa e un nome shadow. Vedere l'esempio seguente su come aggiornare un'ombra denominata.

```
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" }}}'
```

Invio di payload binari

È possibile inviare un payload binario all'API proxy, fino al servizio AWS IoT. Nell'esempio seguente viene inviato il contenuto della copiaREADME.md associato a questo modulo (trattato come dati binari) perdevice/fooArgomento tramiteapplication/octet-streamTipo di contenuto.

```
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: Eseguire questo comando mentre si trova nella directory di questo progetto. È quindi possibile testare l'invio di altri tipi di file binari dal file system.

GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws-solutions-constructs/aws-apigateway-iot](https://github.com/aws-solutions-constructs/aws-apigateway-iot)



aws-apigateway-kinesisstream


STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic Versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | aws_solutions_constructs.aws_apigateway_kinesisstreams |
|  TS | @aws-solutions-constructs/aws-apigateway-kinesisstreams |

| Linguaggio | Pacchetto |
|---|---|
| TypeScript | |
|  Java | software.amazon.awsconstruc ts.services.apigatewaykines isstreams |

Overview

Questo modello implementa un'Amazon API Gateway API connessa a un flusso di dati Amazon Kinesis.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id [string](#)
- props [ApiGatewayToKinesisStreamsProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------------|----------------------------------|---|
| ApiGatewayProps? | api.RestApiProps | Accessori di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per l'API REST del gateway API. |
| PutRecordRequestTemplate? | string | Modello di richiesta API Gateway per l'azione PutRecord. Se non viene fornito, verrà utilizzato uno predefinito. |
| PutRecordRequestModel? | api.ModelOptions | Modello di richiesta API Gateway per l'azione PutRecord. Se non viene fornito, ne verrà creato uno predefinito. |
| PutRecordsRequestTemplate? | string | Modello di richiesta API Gateway per l'azione PutRecords. Se non viene fornito, verrà utilizzato uno predefinito. |
| PutRecordRequestModel? | api.ModelOptions | Modello di richiesta API Gateway per l'azione PutRecords. Se non viene fornito, ne verrà creato uno predefinito. |
| StreamObj esistente? | kinesis.Stream | Istanza esistente di Kinesis Stream, fornendo sia questo che <code>kinesisStreamProps</code> |

| Nome | Tipo | Descrizione |
|---------------------|--|--|
| | | In caso contrario, si verifica un errore |
| KinesisStreamProps? | <u>kinesis.StreamProps</u> | Puntelli opzionali forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il flusso Kinesis. |
| LogGroupProps? | <u>logs.LogGroupProps</u> | Opzionali forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|--------------------------|--------------------------------------|--|
| ApiGateway | <u>api.RestApi</u> | Restituisce un'istanza dell'API REST del gateway API creata dal modello. |
| ApiGatewayRole | <u>iam.Role</u> | Restituisce un'istanza del ruolo IAM creato dal pattern per l'API REST del gateway API. |
| APigatewayCloudWatchRole | <u>iam.Role</u> | Restituisce un'istanza del ruolo IAM creato dal pattern che abilita la registrazione degli accessi dall'API REST del gateway API a CloudWatch. |
| ApiGatewayLogGroup | <u>logs.LogGroup</u> | Restituisce un'istanza del gruppo di log creato dal |

| Nome | Tipo | Descrizione |
|---------------|--------------------------------|--|
| | | modello a cui vengono inviati i log di accesso API REST del API Gateway API. |
| KinesisStream | kinesis.Stream | Restituisce un'istanza del flusso Kinesis creato dal pattern. |

Uso dell'API di esempio

| Metodo | Percorso della richiesta | Corpo della richiesta | Azione coda | Descrizione |
|--------|--------------------------|--|--------------------|---|
| POST | /record | <pre>{ "data": "Hello World!", "partitionKey": "pk001" }</pre> | kinesis:PutRecord | Scrive un singolo record di dati nel flusso. |
| POST | /records | <pre>{ "records": [{ "data": "abc", "partitionKey": "pk001" }, { "data": "xyz",</pre> | kinesis:PutRecords | Scrive più record di dati nel flusso in una singola chiamata. |

| Metodo | Percorso della richiesta | Corpo della richiesta | Azione coda | Descrizione |
|--------|--------------------------|--|-------------|-------------|
| | | <pre> "partitionKey": "pk001" }] } </pre> | | |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

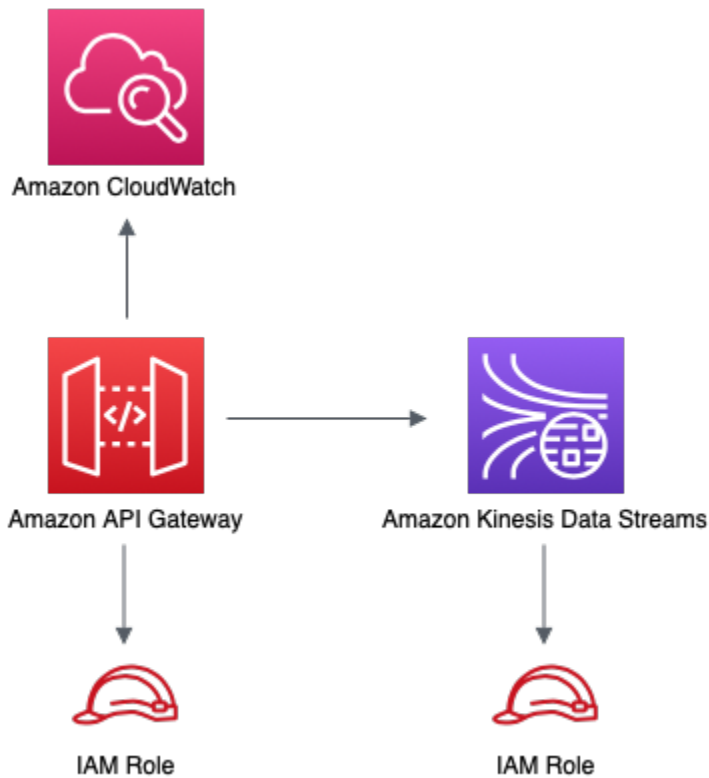
Amazon API Gateway

- Distribuzione di un endpoint API ottimizzato per gli edge.
- Abilitare la registrazione CloudWatch per API Gateway.
- Configurare il ruolo IAM di accesso ai privilegi minimi per il API Gateway.
- Impostare l'AuthorizationType predefinito per tutti i metodi API su IAM.
- Abilitare il tracciamento X-Ray.
- Convalidare il corpo della richiesta prima di passare i dati a Kinesis.

Amazon Kinesis Data Stream

- Configurare il ruolo IAM di accesso ai privilegi minimi per il flusso Kinesis.
- Abilita la crittografia lato server per Kinesis Stream utilizzando AWS Managed KMS Key.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-constructs/aws-apigateway-kinesisstream](https://github.com/aws-solutions-constructs/aws-apigateway-kinesisstream)

aws-apigateway-lambda

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Controllo delle versioni semantiche](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere

necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_apigateway_lambda</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-apigateway-lambda</code> |
|  Java | <code>software.amazon.awsconstructs.services.apigatewaylambda</code> |

Overview

Questo Construct di soluzioni AWS implementa un'Amazon API Gateway API connessa a una funzione AWS Lambda.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id `string`
- props [ApiGatewayToLambdaProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--|--|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| ApiGatewayProps? | api.LambdaRestApiProps | Oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per l'API. |
| LogGroupProps? | logs.LogGroupProps | Gli oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena |

| Nome | Tipo | Descrizione |
|------|------|---|
| | | predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|--------------------------|-----------------------------------|--|
| APigatewayCloudWatchRole | iam.Role | Restituisce un'istanza del ruolo IAM creato dal pattern che abilita la registrazione degli accessi dall'API REST del gateway API a CloudWatch. |
| ApiGatewayLogGroup | logs.LogGroup | Restituisce un'istanza del gruppo di log creato dal modello a cui vengono inviati i log di accesso API REST del API Gateway API. |
| LambdaFunction | lambda.Function | Restituisce un'istanza della funzione Lambda creata dal pattern. |
| ApiGateway | api.LambdaRestApi | Restituisce un'istanza dell'API REST del gateway API creata dal modello. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

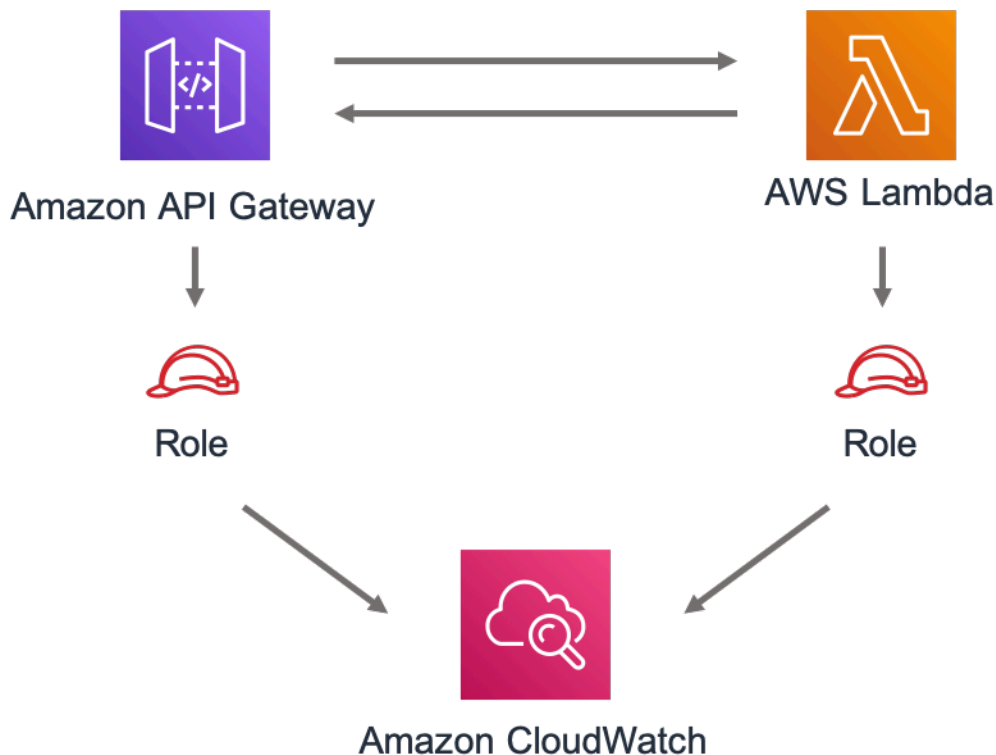
Amazon API Gateway

- Distribuzione di un endpoint API ottimizzato per gli edge
- Attivazione della registrazione CloudWatch per API Gateway
- Configurare il ruolo IAM di accesso ai privilegi minimi per il API Gateway
- Impostare l'AuthorizationType predefinito per tutti i metodi API su IAM
- Abilita il monitoraggio di X-Ray
- Impostare le variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso limitato per la funzione Lambda
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda
- Abilita il monitoraggio di X-Ray

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:






[@aws -solutions-buils/aws-apigateway-lambda](#)

aws-apigateway-sagemakerendpoint

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Versionamento semantico](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_apigateway_sagemakerendpoint</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-apigateway-sagemakerendpoint</code> |
|  Java | <code>software.amazon.awsconstructs.services.apigatewaysagemakerendpoint</code> |

Overview

Questo Construct di soluzioni AWS implementa un'Amazon API Gateway API connessa a un endpoint Amazon SageMaker.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- `scopeConstruct`
- `idstring`
- `propsApiGatewayToSageMakerEndpointProps`

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|---------------------------------------|---|--|
| <code>ApiGatewayProps?</code> | <code>api.RestApiProps</code> | Oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per l'API REST del gateway API. |
| <code>ApiGatewayExecutionRole?</code> | <code>iam.Role</code> | Ruolo IAM utilizzato da API Gateway per richiamare l'endpoint SageMaker. Se non viene specificato, viene creato un ruolo predefinito con accesso a <code>endpointName</code> . |
| <code>EndpointName</code> | <code>string</code> | Nome dell'endpoint di inferenza SageMaker distribuito. |
| <code>ResourceName?</code> | <code>string</code> | Nome risorsa facoltativo in cui sarà disponibile il metodo GET. |
| <code>resourcePath</code> | <code>string</code> | Percorso della risorsa per il metodo GET. La variabile definita qui può essere referenziata in <code>requestMappingTemplate</code> . |
| <code>RequestMappingTemplate</code> | <code>string</code> | Modello di mapping per convertire le richieste GET |

| Nome | Tipo | Descrizione |
|--------------------------|---|--|
| | | ricevute nell'API REST in richieste POST previste dall'endpoint SageMaker. |
| ResponseMappingTemplate? | string | Modello di mapping facoltativo per convertire le risposte ricevute dall'endpoint SageMaker. |
| LogGroupProps? | <u>logs.LogGroupProps</u> | Elementi di scena opzionali forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|--------------------------|--|--|
| ApiGateway | <u>api.LambdaRestApi</u> | Restituisce un'istanza dell'API REST del gateway API creata dal modello. |
| ApiGatewayRole | <u>iam.Role</u> | Restituisce un'istanza del ruolo IAM creato dal pattern per l'API REST del gateway API. |
| APigatewayCloudWatchRole | <u>iam.Role</u> | Restituisce un'istanza del ruolo IAM creato dal pattern che abilita la registrazione degli accessi dall'API REST del gateway API a CloudWatch. |

| Nome | Tipo | Descrizione |
|--------------------|-------------------------------|--|
| ApiGatewayLogGroup | logs.LogGroup | Restituisce un'istanza del gruppo di log creato dal modello a cui vengono inviati i log di accesso API REST del API Gateway API. |

Uso di API di esempio

Nota: Ogni endpoint SageMaker è univoco e la risposta dell'API dipenderà dal modello distribuito. L'esempio riportato di seguito presuppone che il campione da [Post di blog](#): . Per un riferimento su come sarebbe implementato, fare riferimento a [integ.apigateway-sagemakerendpoint-override.ts](#): .

| Metodo | Percorso della richiesta | Stringa di query | Azione SageMaker | Descrizione |
|--------|--------------------------|-------------------|--------------------------|--|
| GET | /321 | items=101,131,162 | sagemaker:InvokeEndpoint | Recupera le previsioni per un utente e gli elementi specifici. |

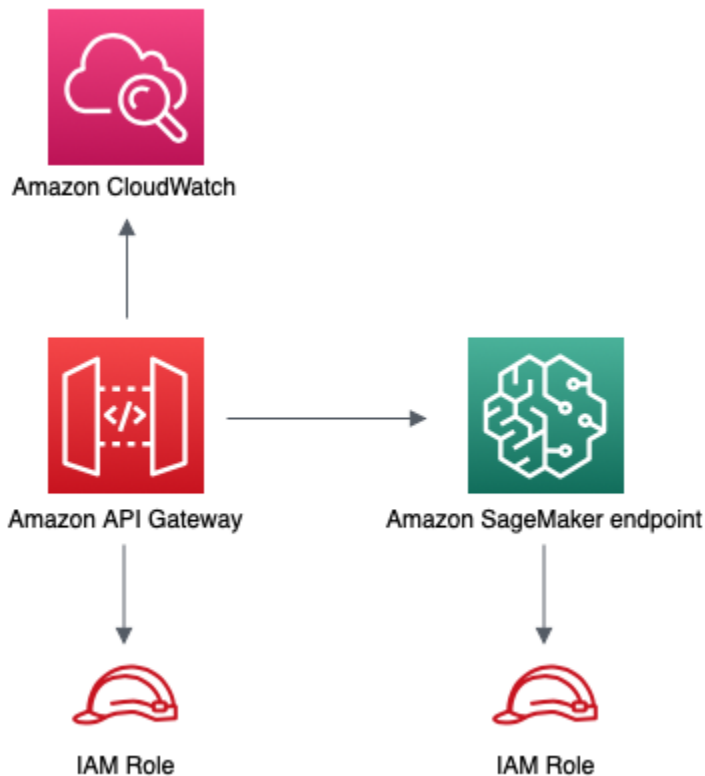
Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Amazon API Gateway

- Distribuzione di un endpoint API ottimizzato per gli edge
- Abilitazione della registrazione CloudWatch per API Gateway
- Configurare il ruolo IAM di accesso ai privilegi minimi per il API Gateway
- Impostare l'AuthorizationType predefinito per tutti i metodi API su IAM
- Abilita la traccia X-Ray
- Convalidare i parametri di richiesta prima di passare i dati a SageMaker

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-constructs/aws-apigateway-sagemakerendpoint](https://github.com/aws-solutions-constructs/aws-apigateway-sagemakerendpoint)


aws-apigateway-sqs

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario

aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_apigateway_sqs</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-apigateway-sqs</code> |
|  Java | <code>software.amazon.awsconstructs.services.apigatewaysqs</code> |

Overview

Questo Construct di soluzioni AWS implementa un'Amazon API Gateway API connessa a una coda Amazon SQS.

Ecco una definizione di modello distribuibile minima in TypeScript:

```
import { ApiGatewayToSqs, ApiGatewayToSqsProps } from "@aws-solutions-constructs/aws-apigateway-sqs";  
  
new ApiGatewayToSqs(this, 'ApiGatewayToSqsPattern', {});
```

Initializer

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

Parametri

- scope [Construct](#)
- idstring
- props [ApiGatewayToSqsProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|------------------------|----------------------------------|--|
| ApiGatewayProps? | api.RestApiProps | Elementi di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il API Gateway. |
| QueueProps? | sqs.QueueProps | Elementi di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per la coda. |
| DeployDeadLetterQueue? | boolean | Indica se distribuire una coda secondaria da utilizzare come coda DLQ. L'impostazione predefinita è <code>true</code> . |
| MaxReceiveCount | number | Il numero di volte in cui un messaggio può essere rimosso dalla coda senza successo prima di essere spostato nella coda DLQ. |
| AllowCreateOperation? | boolean | Se distribuire un metodo API Gateway per le operazioni di creazione sulla coda (ad esempio SQS: <code>SendMessage</code>). |

| Nome | Tipo | Descrizione |
|------------------------|------------------------------------|---|
| CreateRequestTemplate? | string | Sostituisci il modello di richiesta API Gateway predefinito per il metodo Create, se <code>allowCreateOperation</code> è impostato su <code>.true:</code> . |
| AllowReadOperation? | boolean | Se distribuire un metodo API Gateway per le operazioni di lettura nella coda (ad esempio SQS: <code>ReceiveMessage</code>). |
| ReadRequestTemplate? | string | Sostituisci il modello di richiesta API Gateway predefinito per il metodo Read, se <code>allowReadOperation</code> è impostato su <code>.true:</code> . |
| AllowDeleteOperation? | boolean | Se distribuire un metodo API Gateway per le operazioni di eliminazione sulla coda (ad esempio SQS: <code>DeleteMessage</code>). |
| DeleteRequestTemplate? | string | Sostituisci il modello di richiesta API Gateway predefinito per il metodo Delete, se <code>allowDeleteOperation</code> è impostato su <code>.true:</code> . |
| LogGroupProps? | logs.LogGroupProps | Elementi di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|--------------------------|-------------------------------|--|
| ApiGateway | api.RestApi | Restituisce un'istanza dell'API REST del gateway API creata dal modello. |
| APiGatewayCloudWatchRole | iam.Role | Restituisce un'istanza del ruolo IAM creato dal pattern che abilita la registrazione degli accessi dall'API REST del gateway API a CloudWatch. |
| ApiGatewayLogGroup | logs.LogGroup | Restituisce un'istanza del gruppo di log creato dal modello a cui vengono inviati i log di accesso API REST del API Gateway API. |
| ApiGatewayRole | iam.Role | Restituisce un'istanza del ruolo IAM creato dal pattern per l'API REST del gateway API. |
| DeadLetterQueue? | sqs.Queue | Restituisce un'istanza della coda di lettere morte creata dal modello, se ne viene distribuita una. |
| SQSqueue | sqs.Queue | Restituisce un'istanza della coda SQS creata dal pattern. |

Uso di API di esempio

| Metodo | Percorso della richiesta | Corpo della richiesta | Azione coda | Descrizione |
|--------|--------------------------------|----------------------------|---------------------|---|
| GET | / | | sqs::ReceiveMessage | Recupera un messaggio dalla coda. |
| POST | / | { "data": "Hello World!" } | sqs::SendMessage | Consegna un messaggio alla coda. |
| DELETE | /message?receiptHandle=[value] | | sqs::DeleteMessage | Elimina un messaggio specificato dalla coda |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

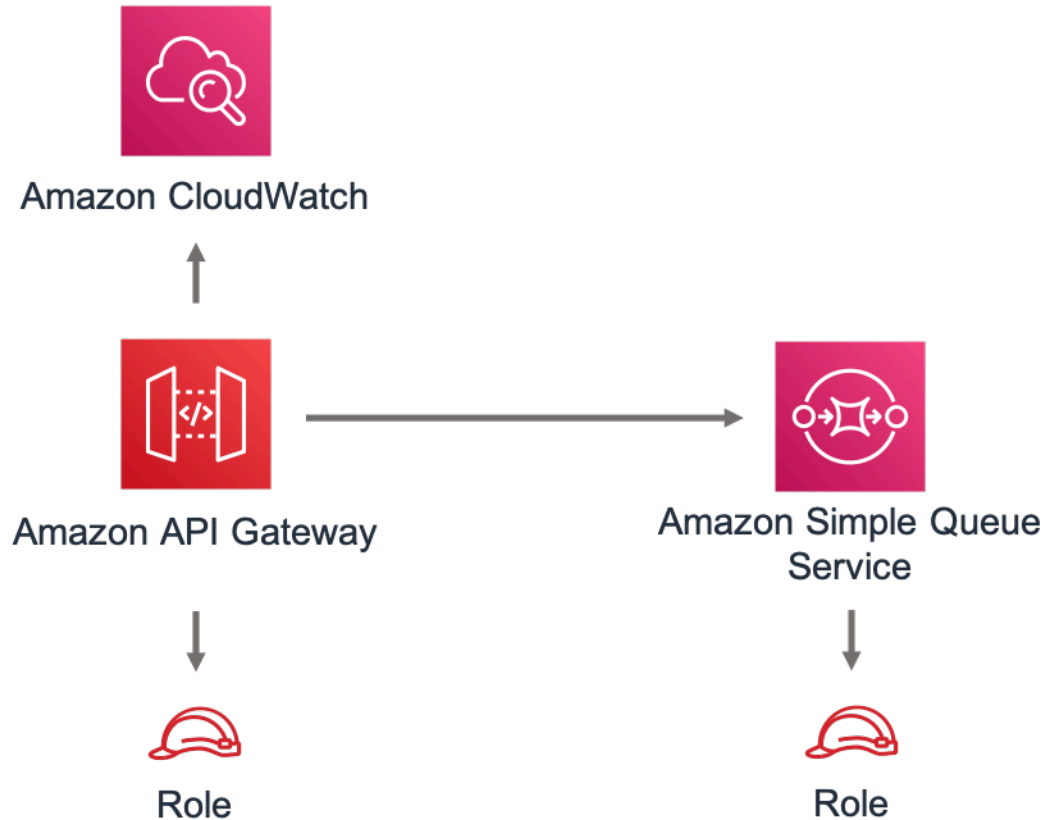
Amazon API Gateway

- Distribuzione di un endpoint API ottimizzato per gli edge
- Abilitare la registrazione CloudWatch per API Gateway
- Configurare il ruolo IAM di accesso ai privilegi minimi per il API Gateway
- Impostare l'AuthorizationType predefinito per tutti i metodi API su IAM
- Abilitare la traccia X-Ray

Coda Amazon SQS

- Distribuzione della coda DLS per la coda SQS di origine
- Abilitare la crittografia lato server per la coda SQS di origine mediante AWS gestita
- Applicazione della crittografia dei dati in transito

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-constructs/aws-apigateway-sqs](https://github.com/aws-solutions-constructs/aws-apigateway-sqs)




aws-cloudfront-apigateway

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic Versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario

aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_cloudfront_apigateway</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-cloudfront-apigateway</code> |
|  Java | <code>software.amazon.awsconstructs.services.cloudfrontapigateway</code> |

Overview

Questo Construct di soluzioni AWS implementa una distribuzione Amazon CloudFront davanti a un'API REST di Amazon API Gateway.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id `string`
- props [CloudFrontToApiGatewayProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|------------------------|-----------------------------|--|
| esistenteApigatewayObj | api.RestApi | Il API Gateway regionale che verrà fronteggiato con CloudFront |

| Nome | Tipo | Descrizione |
|------------------------------|---|--|
| CloudFrontDistributionProps? | <u>cloudfront.DistributionProps</u> | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per la distribuzione CloudFront. |
| InsertHTPTSecurityHeaders? | boolean | L'utente opzionale ha fornito oggetti di scena per attivare/disattivare l'iniezione automatica delle intestazioni di sicurezza HTTP best practice in tutte le risposte da CloudFront |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------------------|---|--|
| ApiGateway | <u>api.RestApi</u> | Restituisce un'istanza dell'API REST del gateway API creata dal modello. |
| CloudFrontLoggingBucket? | <u>s3.Bucket</u> | Restituisce un'istanza del bucket di registrazione creato dal pattern per la distribuzione Web CloudFront. |
| CloudFrontWebDistribution | <u>cloudfront.CloudFrontWebDistribution</u> | Restituisce un'istanza della distribuzione Web CloudFront creata dal pattern. |
| EdgeLambdafunctionVersion? | <u>lambda.Version</u> | Restituisce un'istanza della versione della funzione bordo Lambda creata dalla serie. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

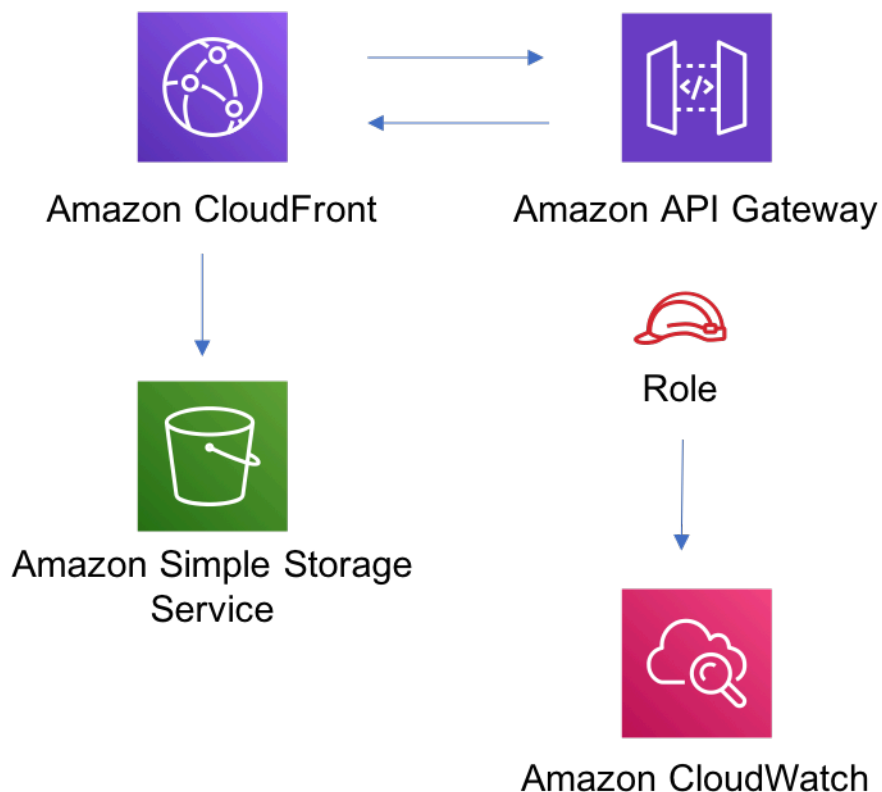
Amazon CloudFront

- Configurazione della registrazione di accesso per CloudFront WebDistribution
- Abilitare l'iniezione automatica delle intestazioni di sicurezza HTTP best practice in tutte le risposte da CloudFront WebDistribution

Amazon API Gateway

- L'oggetto API Gateway fornito dall'utente viene utilizzato così com'è
- Abilita il monitoraggio di X-Ray

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:






[@aws-solutions-constructs/aws-cloudfront-apigateway](https://github.com/aws-solutions-constructs/aws-cloudfront-apigateway)

aws-cloudfront-apigateway-lambda

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti all'[controllo delle versioni semantiche](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | aws_solutions_constructs.aws_s_cloudfront_apigateway_lambda |
|  TypeScript | @aws-solutions-constructs/aws-cloudfront-apigateway-lambda |
|  Java | software.amazon.awsconstructs.services.cloudfrontapigatewaylambda |

Overview

Questo AWS Solutions Construct implementa una distribuzione Amazon CloudFront davanti a un'API REST supportata da Amazon API Gateway Lambda.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- `scope` [Construct](#)
- `id` `string`
- `props` [CloudFrontToApiGatewayToLambdaProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------------------|---------------------------------|--|
| <code>EsistenteLambdaobj?</code> | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo |

| Nome | Tipo | Descrizione |
|---|--|--|
| | | sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| <code>LambdaFunctionProps?</code> | <u><code>lambda.FunctionProps</code></u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| <code>ApiGatewayProps?</code> | <u><code>api.LambdaRestApiProps</code></u> | Gli oggetti di scena forniti dall'utente facoltativi per sovrascrivere gli oggetti di scena predefiniti per API Gateway |
| <code>CloudFrontDistributionProps?</code> | <u><code>cloudfront.DistributionProps</code></u> | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per la distribuzione CloudFront. |
| <code>InsertHTPTSecurityHeaders?</code> | <code>boolean</code> | L'utente opzionale ha fornito oggetti di scena per attivare/disattivare l'iniezione automatica delle intestazioni di sicurezza HTTP best practice in tutte le risposte da CloudFront |
| <code>LogGroupProps?</code> | <u><code>logs.LogGroupProps</code></u> | Elementi di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------------------|---|--|
| ApiGateway | <u>api.RestApi</u> | Restituisce un'istanza dell'API REST del gateway API creata dal modello. |
| APigatewayCloudWatchRole | <u>iam.Role</u> | Restituisce un'istanza del ruolo IAM creato dal pattern che abilita la registrazione degli accessi dall'API REST del gateway API a CloudWatch. |
| ApiGatewayLogGroup | <u>logs.LogGroup</u> | Restituisce un'istanza del gruppo di log creato dal modello a cui vengono inviati i log di accesso API REST del API Gateway API. |
| CloudFrontLoggingBucket? | <u>s3.Bucket</u> | Restituisce un'istanza del bucket di registrazione creato dal pattern per la distribuzione Web CloudFront. |
| CloudFrontWebDistribution | <u>cloudfront.CloudFrontWebDistribution</u> | Restituisce un'istanza della distribuzione Web CloudFront creata dal pattern. |
| EdgeLambdafunctionVersion? | <u>lambda.Version</u> | Restituisce un'istanza della versione della funzione Lambda creata dalla serie. |
| LambdaFunction | <u>lambda.Function</u> | Restituisce un'istanza della funzione Lambda creata dal pattern. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Amazon CloudFront

- Configurazione della registrazione di accesso per CloudFront WebDistribution
- Abilitare l'iniezione automatica delle intestazioni di sicurezza HTTP best practice in tutte le risposte da CloudFront WebDistribution

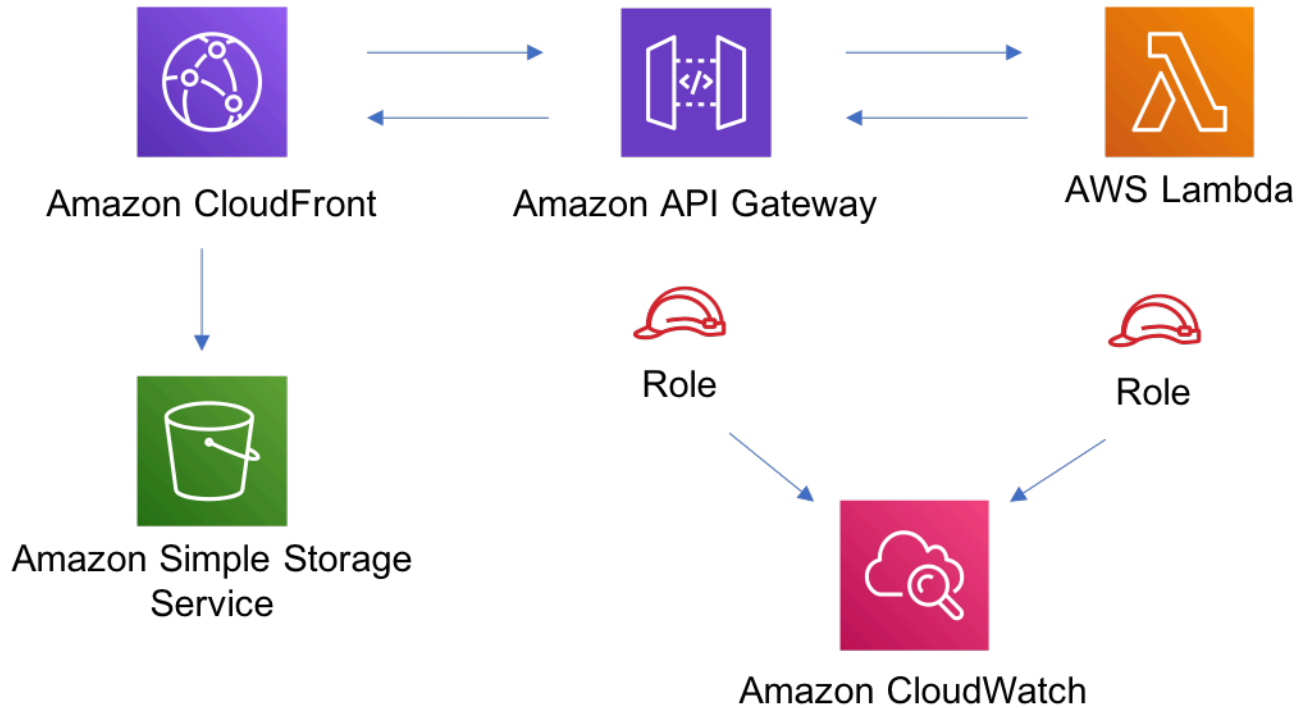
Amazon API Gateway

- Distribuire un endpoint API regionale
- Abilita la registrazione di CloudWatch per API Gateway
- Configurare il ruolo IAM di accesso ai privilegi minimi per il API Gateway
- Impostare l'AuthorizationType predefinito per tutti i metodi API su IAM
- Abilita il monitoraggio di X-Ray

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso limitato per la funzione Lambda
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda
- Abilita il monitoraggio di X-Ray
- Impostare le variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:






[@aws -solutions-constructs/aws-cloudfront-apigateway-lambda](https://github.com/aws-solutions-constructs/aws-cloudfront-apigateway-lambda)

aws-cloudfront-mediastore

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Versionamento semantico](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_cloudfront_mediastore</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-cloudfront-mediastore</code> |
|  Java | <code>software.amazon.awsconstructs.services.cloudfrontmediastore</code> |

Overview

Questo AWS Solutions Construct implementa una distribuzione Amazon CloudFront collegata a un contenitore AWS Elemental MediaStore.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- [scopeConstruct](#)
- `idstring`
- [propsCloudFrontToMediaStoreProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------------------|--|---|
| EsistenteMediaStorecontainerObj? | mediastore.CfnContainer | Contenitore MediaStore facoltativo fornito dall'utente per sovrascrivere il contenitore MediaStore predefinito. |
| MediaStorecontainerProps? | mediastore.CfnContainerProps | Elementi di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il contenitore MediaStore. |
| CloudFrontDistributionProps? | cloudfront.DistributionProps any | Elementi di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per la distribuzione CloudFront. |
| InsertHTPTSecurityHeaders? | boolean | Puntelli opzionali forniti dall'utente per attivare/disattivare l'iniezione automatica delle intestazioni di sicurezza HTTP best practice in tutte le risposte da CloudFront. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|---------------------------------|--|--|
| CloudFrontWebDistribution | cloudfront.CloudFrontWebDistribution | Restituisce un'istanza della distribuzione Web CloudFront creata dal pattern. |
| MediaStoreContainer | mediastore.CfnContainer | Restituisce un'istanza del contenitore MediaStore creato dal pattern. |
| CloudFrontLoggingBucket | s3.Bucket | Restituisce un'istanza del bucket di registrazione creato dal pattern per la distribuzione Web CloudFront. |
| CloudFrontOriginRequestPolicy | cloudfront.OriginRequestPolicy | Restituisce un'istanza del criterio di richiesta di origine CloudFront creato dal pattern per la distribuzione Web CloudFront. |
| CloudFrontOriginAccessIdentity? | cloudfront.OriginAccessIdentity | Restituisce un'istanza dell'identità di accesso all'origine CloudFront creata dal pattern per la distribuzione Web CloudFront. |
| EdgeLambdafunctionVersion | lambda.Version | Restituisce un'istanza della versione della funzione bordo Lambda creata dalla serie. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

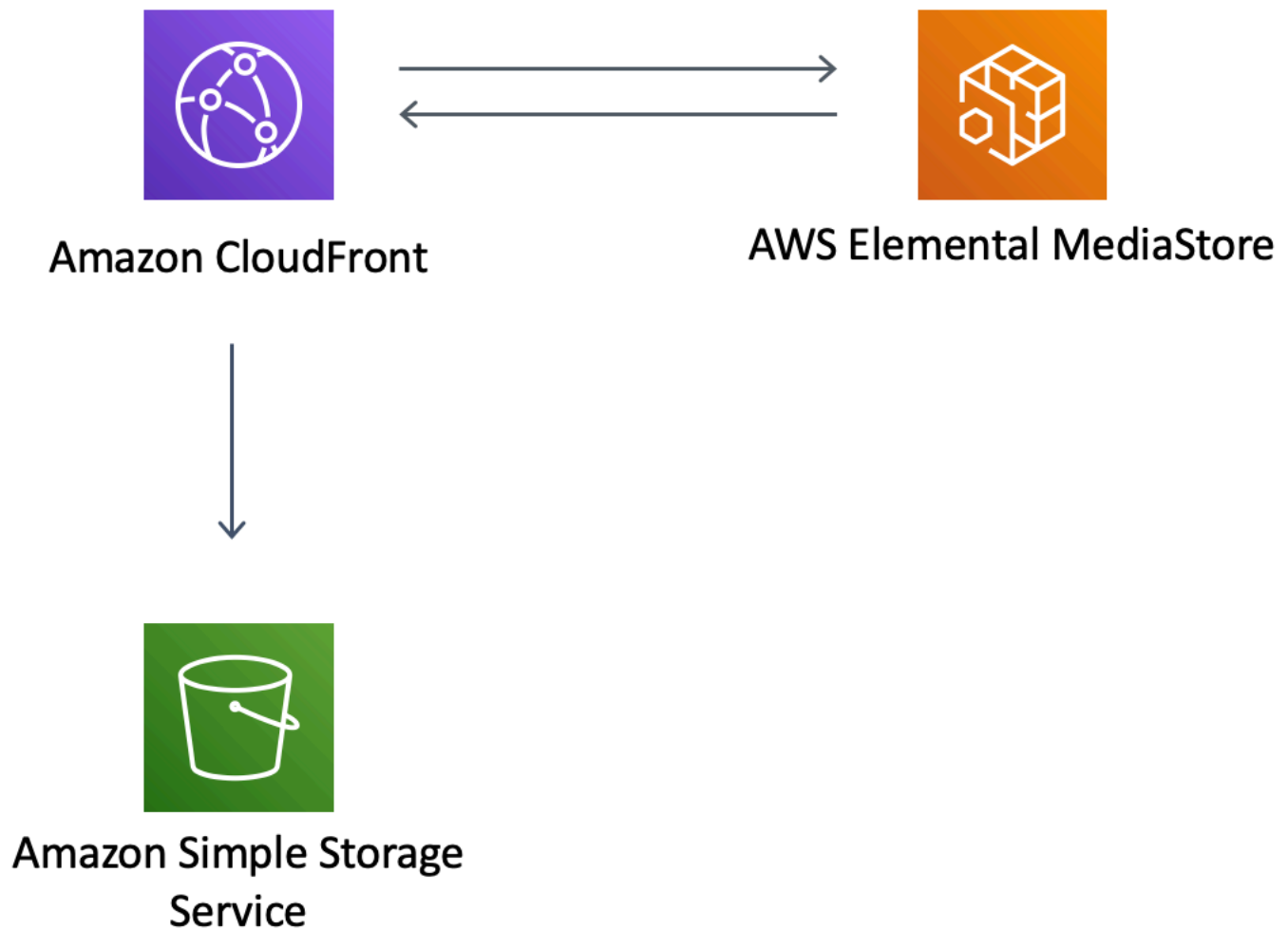
Amazon CloudFront

- Configurare la registrazione degli accessi per la distribuzione CloudFront
- Abilitare i criteri di richiesta di origine CloudFront per il contenitore AWS Elemental MediaStore
- Imposta `User-Agent` intestazione personalizzata con identità di accesso origine di CloudFront
- Abilitare l'iniezione automatica delle intestazioni di sicurezza HTTP best practice in tutte le risposte dalla distribuzione web CloudFront

AWS Elemental MediaStore

- Impostare il criterio di eliminazione per mantenere la risorsa
- Impostare il nome del contenitore con il nome dello stack CloudFormation
- Imposta il valore di default [Policy CORS \(Cross-Origin Resource Sharing\) di](#)
- Imposta il valore di default [Policy del ciclo di vita degli oggetti](#)
- Imposta il valore di default [Policy di container](#) per consentire solo `aws:UserAgent` con identità di accesso origine di CloudFront
- Imposta il valore di default [Policy di parametro](#)
- Abilitazione della registrazione degli accessi

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-constructs/aws-cloudfront-mediastore](https://github.com/aws-solutions-constructs/aws-cloudfront-mediastore)



aws-cloudfront-s3

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_s3_cloudfront_s3</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-cloudfront-s3</code> |
|  Java | <code>software.amazon.awsconstructs.services.cloudfronts3</code> |

Overview

Questo Construct di soluzioni AWS implementa una distribuzione Amazon CloudFront davanti a un bucket Amazon S3.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- `scope` [Construct](#)
- `id` `string`
- `props` [CloudFrontToS3Props](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|------------------------------|--|---|
| Esistente BucketObj? | s3.Bucket | Istanza esistente dell'oggetto S3 Bucket. Se questo è fornito, quindi anche <code>provideBucketProps</code> è un errore. |
| Bucket Props? | s3.BucketProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per il bucket. Ignorato se <code>unexistingBucketObj</code> viene fornito. |
| CloudFrontDistributionProps? | cloudfront.DistributionProps | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per la distribuzione CloudFront. |
| InsertHTPTSecurityHeaders? | <code>boolean</code> | L'utente opzionale ha fornito oggetti di scena per attivare/disattivare l'iniezione automatica delle intestazioni di sicurezza HTTP best |

| Nome | Tipo | Descrizione |
|------|------|---|
| | | practice in tutte le risposte da CloudFront |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------------------|--|--|
| CloudFrontWebDistribution | cloudfront.CloudFrontWebDistribution | Restituisce un'istanza della distribuzione Web CloudFront creata dal pattern. |
| S3bucket? | s3.Bucket | Restituisce un'istanza del bucket S3 creato dalla serie. |
| S3loggingBucket? | s3.Bucket | Restituisce un'istanza del bucket di registrazione creato dal pattern per il bucket S3. |
| EdgeLambdafunctionVersion? | lambda.Version | Restituisce un'istanza della versione della funzione bordo Lambda creata dalla serie. |
| CloudFrontLoggingBucket? | s3.Bucket | Restituisce un'istanza del bucket di registrazione creato dal pattern per la distribuzione Web CloudFront. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Amazon CloudFront

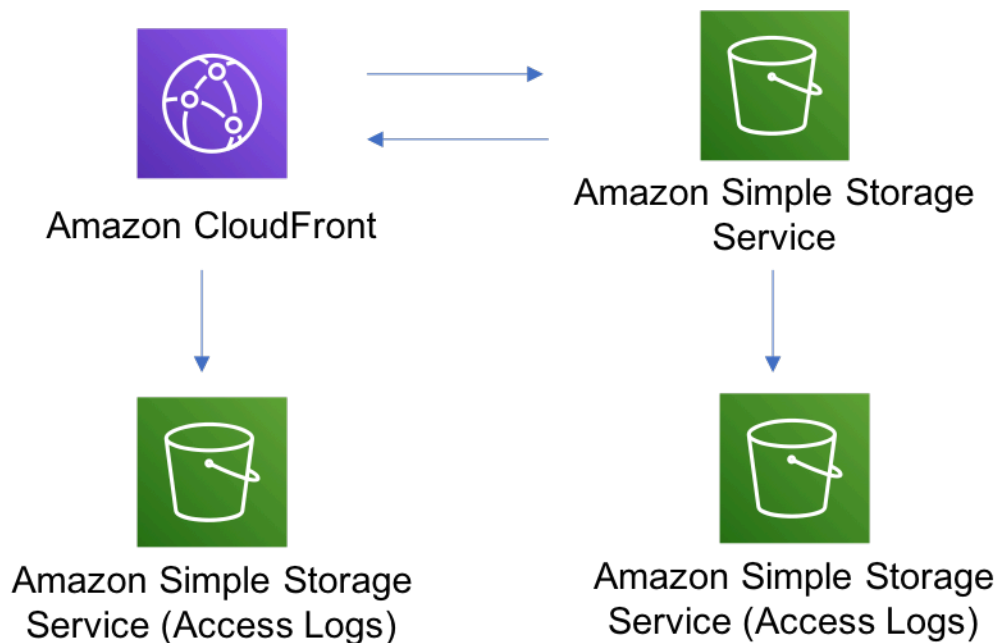
- Configurazione della registrazione di accesso per CloudFront WebDistribution

- Abilitare l'iniezione automatica delle intestazioni di sicurezza HTTP best practice in tutte le risposte da CloudFront WebDistribution

Amazon S3

- Configurare la registrazione degli accessi per S3
- Abilita la crittografia lato server per S3 Bucket utilizzando la chiave AWS gestita
- Attivare il controllo delle versioni per S3 Bucket
- Non consentire l'accesso pubblico per S3 Bucket
- Mantenere il bucket S3 quando si elimina lo stack CloudFormation
- Applicazione della crittografia dei dati in transito
- Applica la regola del ciclo di vita per spostare versioni degli oggetti non correnti nell'archivio Glacier dopo 90 giorni

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-constructs/aws-cloudfront-s3](#)




aws-cognito-apigateway-lambda

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_cognito_apigateway_lambda</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-cognito-apigateway-lambda</code> |
|  Java | <code>software.amazon.awsconstructs.services.cognitoapigatewaylambda</code> |

Overview

Questo Construct di soluzioni AWS implementa Amazon Cognito proteggendo un'API REST supportata da Amazon API Gateway Lambda.

Ecco una definizione di modello distribuibile minima in 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'
  }
});
```

Se stai definendo risorse e metodi sulla tua API (ad es. `proxy = false`), è necessario chiamare `iladdAuthorizers()` dopo che l'API è completamente definita. Ciò garantisce che ogni metodo nella tua API sia protetto.

Di seguito è illustrato un esempio in 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);
```

Parametri

- scope [Construct](#)
- id [string](#)
- props [CognitoToApiGatewayToLambdaProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--------------------------------------|--|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |

| Nome | Tipo | Descrizione |
|-----------------------------|---|---|
| ApiGatewayProps? | <u>api.LambdaRestApiProps</u> | Gli oggetti di scena forniti dall'utente facoltativi per sovrascrivere gli oggetti di scena predefiniti per API Gateway |
| CognitouserPoolProps? | <u>cognito.UserPoolPr ops</u> | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per il pool di utenti di Cog |
| CognitouserPoolClientProps? | <u>cognito.UserPoolCl ientProps</u> | Gli oggetti di scena forniti dall'utente facoltativo per sovrascrivere gli oggetti di scena predefiniti per il client del pool di utenti di Cog |
| LogGroupProps? | <u>logs.LogGroupProps</u> | Opzionali degli oggetti di scena forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|------------|------------------------------------|--|
| ApiGateway | <u>api.RestApi</u> | Restituisce un'istanza dell'API REST del gateway API creata dal modello. |

| Nome | Tipo | Descrizione |
|--------------------------|--|--|
| LambdaFunction | lambda.Function | Restituisce un'istanza della funzione Lambda creata dal pattern. |
| userPool | cognito.UserPool | Restituisce un'istanza del pool di utenti Cognito creato dal pattern. |
| UserPoolclient | cognito.UserPoolClient | Restituisce un'istanza del client del pool di utenti Cognito creato dal pattern. |
| APIgatewayCloudWatchRole | iam.Role | Restituisce un'istanza del ruolo IAM creato dal pattern che abilita la registrazione degli accessi dall'API REST del gateway API a CloudWatch. |
| ApiGatewayLogGroup | logs.LogGroup | Restituisce un'istanza del gruppo di log creato dal modello a cui vengono inviati i log di accesso API REST del API Gateway API. |
| APIgatewayAuthorizer | api.CfnAuthorizer | Restituisce un'istanza dell'autorizzazione API Gateway creata dal modello. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Amazon Cognito

- Impostazione dei criteri per le password di utenti di.
- Applica la modalità di protezione avanzata per i pool di utenti

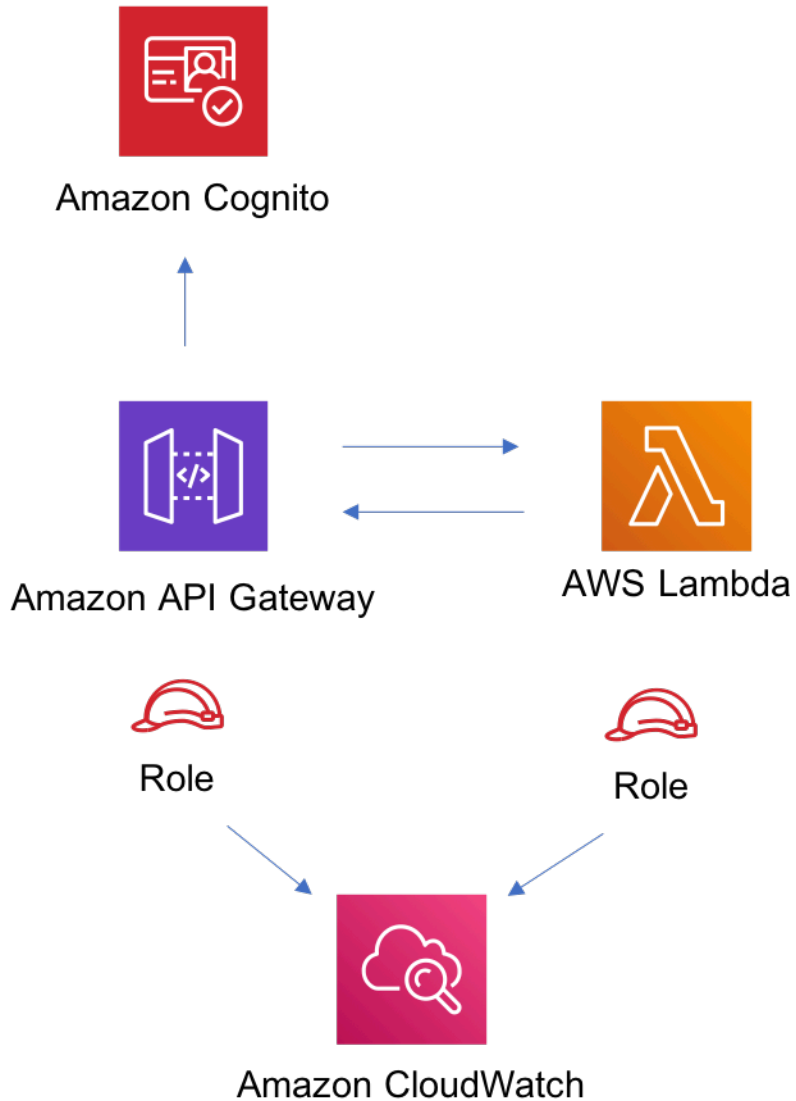
Amazon API Gateway

- Distribuzione di un endpoint API ottimizzato per gli edge
- Attivazione della registrazione CloudWatch per API Gateway
- Configurare il ruolo IAM di accesso ai privilegi minimi per il API Gateway
- Impostare l'AuthorizationType predefinito per tutti i metodi API su IAM
- Abilita il monitoraggio di X-Ray

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda
- Abilita il monitoraggio di X-Ray
- Impostazione delle variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -soluzioni-costruzioni/aws-cognito-api-gateway-lambda](https://github.com/@aws-soluzioni-costruzioni/aws-cognito-api-gateway-lambda)




AWS-Dynamodb-stream-lambda

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic versioning \(Version\) Modello](#). Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|--|--|
|  Python | <code>aws_solutions_constructs.aws_dynamodb_stream_lambda</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-dynamodb-stream-lambda</code> |
|  Java | <code>software.amazon.awsconstructs.services.dynamodbstreamlambda</code> |

Overview

Questo AWS Solutions Construct implementa un modello di tabella Amazon DynamoDB con flusso per richiamare la funzione AWS Lambda con le autorizzazioni meno privilegiate.

Ecco una definizione minima del modello distribuibile:

```
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);

```

Parametri

- scope [Construct](#)
- id [string](#)
- props [DynamoDBStreamToLambdaProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--------------------------------------|--|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |

| Nome | Tipo | Descrizione |
|-------------------------|---|---|
| DynamoTableProps? | dynamodb.TableProps | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per DynamoDB Table |
| TableObj esistente? | dynamodb.Table | Istanza esistente dell'oggetto tabella DynamoDB, fornendo sia questo che <code>dynamodb.TableProps</code> causerà un errore. |
| DynamoEventSourceProps? | aws-lambda-event-sources.DynamoEventSourceProps | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per l'origine eventi DynamoDB |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------|---------------------------------|---|
| DynamoTable | dynamodb.Table | Restituisce un'istanza della tabella DynamoDB creata dalla serie. |
| LambdaFunction | lambda.Function | Restituisce un'istanza della funzione Lambda creata dal pattern. |

valida e completa

Questo modello richiede una funzione Lambda in grado di pubblicare dati nel servizio Elasticsearch dal flusso DynamoDB. Viene fornita una funzione di esempio [Qui](#): .

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

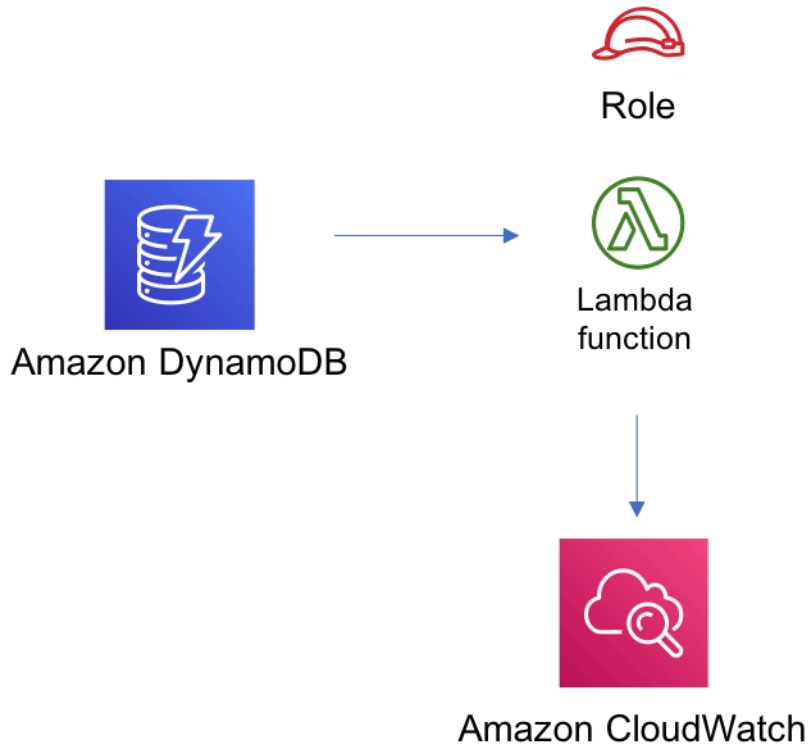
Tabella di Amazon DynamoDB

- Impostare la modalità di fatturazione per DynamoDB Table su richiesta (Paga per richiesta)
- Abilitare la crittografia lato server per la tabella DynamoDB utilizzando la chiave AWS gestita KMS
- Crea una chiave di partizione chiamata 'id' per DynamoDB Table
- Mantieni la tabella quando si elimina lo stack CloudFormation
- Backup continui e ripristino point-in-time

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso limitato per la funzione Lambda
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda
- Abilita il tracciamento X-Ray
- Abilita funzionalità di gestione degli errori: abilita la bisezione sulla funzione Errore; imposta la durata massima record predefinita (24 ore); imposta i tentativi di tentativi massimi predefiniti (500) e distribuisci la coda di lettere non recapitate SQS come destinazione in caso di errore
- Impostare le variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:






[@aws -solutions-buils/aws-dynamodb-stream-lambda](#)

aws-dynamodb-stream-lambda-elasticsearch-kibana

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti all'[controllo semantico delle versioni](#)Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | aws_solutions_constructs.aws_dynamodb_stream_lambda_elasticsearch_kibana |
|  TypeScript | @aws-solutions-constructs/aws-dynamodb-stream-lambda-elasticsearch-kibana |
|  Java | software.amazon.awsconstructs.services.dynamodbstreamlambdaelasticsearchkibana |

Overview

Questo AWS Solutions Construct implementa la tabella Amazon DynamoDB con stream, una funzione AWS Lambda e un Amazon Elasticsearch Service con le autorizzazioni meno privilegiate.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id string
- props [DynamoDBStreamToLambdaToElasticSearchAndKibanaProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--------------------------------------|--|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| DynamoTableProps? | dynamodb.TableProps | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere |

| Nome | Tipo | Descrizione |
|-------------------------|--|---|
| | | vere gli oggetti di scena predefiniti per DynamoDB Table |
| TableObj esistente? | <u>dynamodb.Table</u> | Istanza esistente dell'oggetto tabella DynamoDB, fornendo sia questo che dynamoTableProps causerà un errore. |
| DynamoEventSourceProps? | <u>aws-lambda-event-sources.DynamoEventSourceProps</u> | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per l'origine eventi DynamoDB |
| EsDomainProps? | <u>elasticsearch.CfnDomainProps</u> | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per il Amazon Elasticsearch Service |
| domainName | string | Nome di dominio per il servizio Cognito e Amazon Elasticsearch |
| CreateCloudWatchalarms | boolean | Indica se creare allarmi CloudWatch consigliati. |

Proprietà pattern

| Nome | Tipo | Descrizione |
|-------------------|---|--|
| Cloudwatchalarms? | <u>cloudwatch.Alarm[]</u> | Restituisce un elenco di uno o più avvisi CloudWatch creati dal pattern. |

| Nome | Tipo | Descrizione |
|---------------------|---|--|
| DynamoTable | dynamodb.Table | Restituisce un'istanza della tabella DynamoDB creata dalla serie. |
| ElasticSearchDomain | elasticsearch.CfnDomain | Restituisce un'istanza del dominio Elasticsearch creato dal pattern. |
| IdentityPool | cognito.CfnIdentityPool | Restituisce un'istanza del pool di identità Cognito creato dal pattern. |
| LambdaFunction | lambda.Function | Restituisce un'istanza della funzione Lambda creata dal pattern. |
| userPool | cognito.UserPool | Restituisce un'istanza del pool di utenti Cognito creato dal pattern. |
| UserpoolClient | cognito.UserPoolClient | Restituisce un'istanza del client del pool di utenti Cognito creato dal pattern. |

valida e completa

Questo modello richiede una funzione Lambda in grado di pubblicare dati nel servizio Elasticsearch dal flusso DynamoDB. Viene fornita una funzione di esempio [Qui](#): .

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Tabella Amazon DynamoDB

- Impostare la modalità di fatturazione per DynamoDB Table su richiesta (Paga per richiesta)

- Abilitare la crittografia lato server per la tabella DynamoDB utilizzando la chiave AWS gestita
- Crea una chiave di partizione chiamata 'id' per DynamoDB Table
- Mantieni la tabella quando si elimina lo stack CloudFormation
- Abilita backup continui e ripristino point-in-time

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso limitato per la funzione Lambda
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda
- Abilita la traccia X-Ray
- Abilita funzionalità di gestione degli errori: abilita la bisezione sulla funzione Errore; imposta la durata massima record predefinita (24 ore); imposta i tentativi di tentativi massimi predefiniti (500) e distribuisci la coda di lettere non recapitate SQS come destinazione in caso di errore
- Impostare le variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

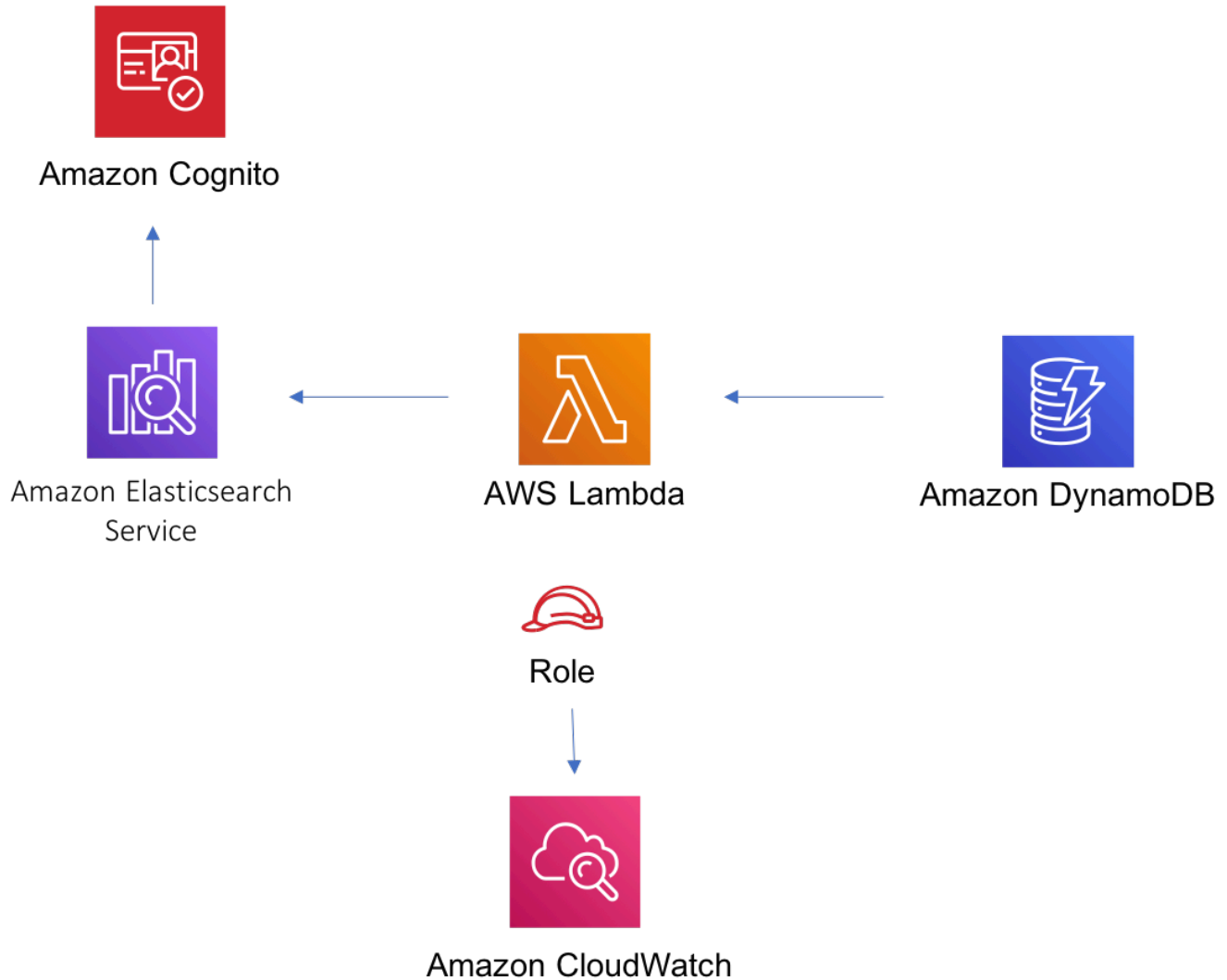
Amazon Cognito

- Impostazione dei criteri delle password per i pool di utenti
- Applica la modalità di protezione avanzata per i pool di utenti

Amazon Elasticsearch Service

- Distribuisci le best practice CloudWatch Allarms per il dominio Elasticsearch
- Proteggi l'accesso alla dashboard di Kibana con i pool di utenti di Cognito
- Abilitare la crittografia lato server per Elasticsearch Domain utilizzando la chiave AWS gestita KMS
- Abilita la crittografia da nodo a nodo per Elasticsearch Domain
- Configurare il cluster per il dominio Amazon ES

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-constructs/aws-dynamodb-stream-lambda-elasticsearch-kibana](https://github.com/aws-solutions-constructs/aws-dynamodb-stream-lambda-elasticsearch-kibana)




aws-eventi-rule-kinesisfirehose-s3

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic Versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | aws_solutions_constructs.aws_events_rule_kinesisfirehose_s3 |
|  TypeScript | @aws-solutions-constructs/aws-events-rule-kinesisfirehose-s3 |
|  Java | software.amazon.awsconstructs.services.eventsrulekinesisfirehoses3 |

Overview

Questo AWS Solutions Construct implementa una regola Amazon CloudWatch Events per inviare i dati a un flusso di consegna Amazon Kinesis Data Firehose collegato a un bucket Amazon S3.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id `string`
- props [EventsRuleToKinesisFirehoseToS3Props](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--|---|
| EventRuleProps | events.RuleProps | Proprietà fornite dall'utente per sovrascrivere le proprietà predefinite per la regola Eventi CloudWatch. |
| KinesisFirehosepuns? | aws-kinesisfirehose.CfnDeliveryStreamProps | L'utente opzionale ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per Kinesis Firehose Delivery Stream. |
| Esistente BucketObj? | s3.IBucket | Istanza esistente dell'oggetto S3 Bucket. Se questo è fornito, quindi anche |

| Nome | Tipo | Descrizione |
|----------------|---|--|
| | | fornirebucketProps è un errore. |
| Bucket Props? | <u>s3.BucketProps</u> | Oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il bucket S3. |
| LogGroupProps? | <u>logs.LogGroupProps</u> | Elementi di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà pattern

| Nome | Tipo | Descrizione |
|------------------|--|---|
| EventsRule | <u>events.Rule</u> | Restituisce un'istanza della regola Events creata dal pattern. |
| KinesisFireHose | <u>kinesisfirehose.CfnDeliveryStream</u> | Restituisce un'istanza del flusso di consegna Kinesis Firehose creato dal pattern. |
| S3Bucket | <u>s3.Bucket</u> | Restituisce un'istanza del bucket S3 creato dalla serie. |
| S3loggingBucket? | <u>s3.Bucket</u> | Restituisce un'istanza del bucket di registrazione creato dal pattern per il bucket S3. |

| Nome | Tipo | Descrizione |
|-------------------------|-------------------------------|--|
| EventsRole? | iam.Role | Restituisce un'istanza del ruolo creato dal costrutto per la regola Eventi CloudWatch. |
| KinesisFireHoserole | iam.Role | Restituisce un'istanza del ruolo IAM creato dal pattern per il flusso di consegna Kinesis Firehose. |
| KinesisFireHoseLogGroup | logs.LogGroup | Restituisce un'istanza del gruppo di log creato dal modello a cui vengono inviati i log di accesso Kinesis Firehose. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Regola Amazon CloudWatch Events

- Configurare il ruolo IAM di accesso ai privilegi minimi per la regola eventi da pubblicare nel flusso di consegna Kinesis Firehose.

Amazon Kinesis Firehose

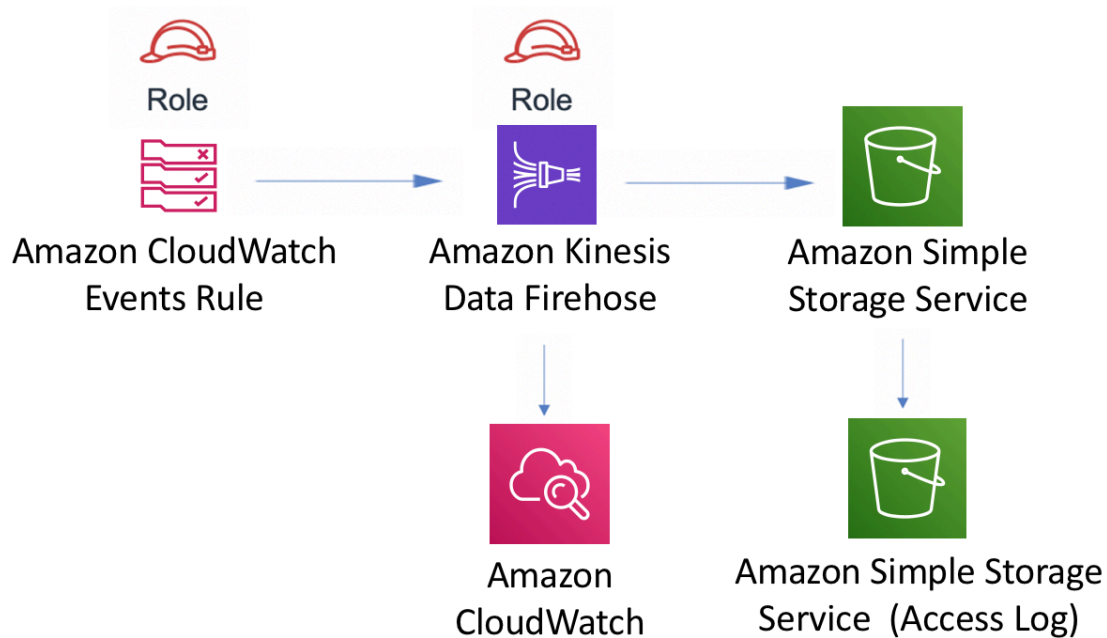
- Abilita la registrazione CloudWatch per Kinesis Firehose.
- Configurare il ruolo IAM di accesso ai privilegi minimi per Amazon Kinesis Firehose.

Benket Amazon S3

- Configurare la registrazione degli accessi per bucket.
- Abilitare la crittografia lato server per il bucket utilizzando la chiave AWS gestita KMS.
- Attivare il controllo delle versioni per il bucket.

- Non consentire l'accesso pubblico per il bucket.
- Mantenere il bucket quando si elimina lo stack CloudFormation.
- Applica la regola del ciclo di vita per spostare le versioni degli oggetti non correnti nell'archivio Glacier dopo 90 giorni.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws-solutions-constructs/aws-events-rule-kinesisfirehose-s3](https://github.com/aws-solutions-constructs/aws-events-rule-kinesisfirehose-s3)



aws-eventi-rule-kinesisstream

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic Versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_events_rule_kinesisstream</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-events-rule-kinesisstreams</code> |
|  Java | <code>software.amazon.awsconstructs.services.eventsrulekinesisstream</code> |

Overview

Questo Construct di soluzioni AWS implementa una regola di Amazon CloudWatch Events per inviare dati a un flusso di dati Amazon Kinesis.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- idstring
- props [EventsRuleToKinesisStreamsProps](#)

Parametri Costrutto

| Nome | Tipo | Descrizione |
|------------------------|-------------------------------------|---|
| EventRuleProps | events.RuleProps | Proprietà fornite dall'utente per sovrascrivere le proprietà predefinite per la regola Eventi CloudWatch. |
| Streamobj esistenteBJ? | kinesis.Stream | Istanza esistente di Kinesis Stream, fornendo sia questo che <code>kinesisStreamProps</code> . In caso contrario, si verifica un errore. |
| KinesisStreamProps? | kinesis.StreamProps | Elementi di scena opzionali forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il flusso Kinesis. |

| Nome | Tipo | Descrizione |
|------------------------|---------|--|
| CreateCloudWatchalarms | boolean | Indica se creare allarmi CloudWatch consigliati. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|---------------|---------------------------------------|--|
| EventsRule | <u>events.Rule</u> | Restituisce un'istanza della regola Events creata dal pattern. |
| KinesisStream | <u>kinesis.Stream</u> | Restituisce un'istanza del flusso Kinesis creato dal pattern. |
| EventsRole? | <u>iam.Role</u> | Restituisce un'istanza del ruolo creato dal costrutto per la regola Eventi CloudWatch. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

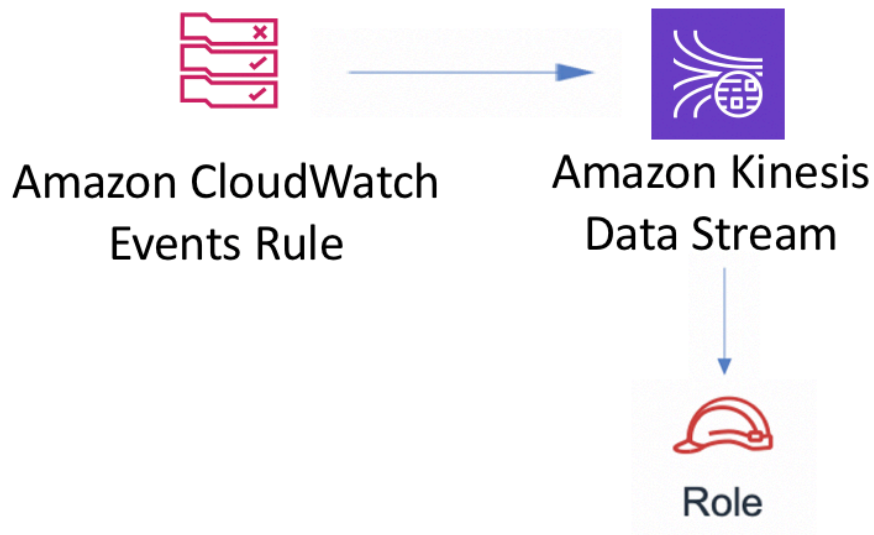
Amazon CloudWatch Events

- Configurare il ruolo IAM di accesso ai privilegi minimi per la regola eventi da pubblicare nel flusso di dati Kinesis.

Amazon Kinesis Stream

- Abilita la crittografia lato server per il flusso di dati Kinesis utilizzando AWS Managed KMS Key.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-constructs/aws-events-rule-kinesisstream](https://github.com/aws-solutions-constructs/aws-events-rule-kinesisstream)

aws-eventi-rule-lambda

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic versioning](#) ([Controllo delle](#)Modello). Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario

aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_events_rule_lambda</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-events-rule-lambda</code> |
|  Java | <code>software.amazon.awsconstructs.services.eventsrulelambda</code> |

Overview

Questo AWS Solutions Construct implementa una regola AWS Events e una funzione AWS Lambda.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id [string](#)
- props [EventsRuleToLambdaProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|---------------------|--------------------------------------|--|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> . In caso contrario, si verifica un errore. |
| LambdafunctionPunps | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| EventRuleProps | events.RuleProps | L'utente ha fornito <code>EventRuleProps</code> per sovrascrivere i valori predefiniti. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------|---------------------------------|--|
| EventsRule | events.Rule | Restituisce un'istanza della regola Events creata dal pattern. |
| LambdaFunction | lambda.Function | Restituisce un'istanza della funzione Lambda creata dal pattern. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

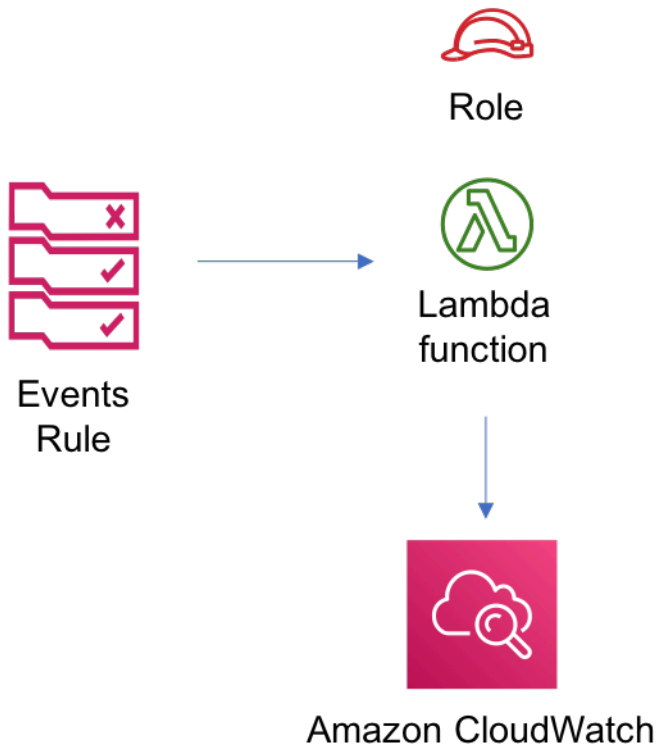
Amazon CloudWatch Events Regola

- Concedere le autorizzazioni di privilegi minimi agli eventi CloudWatch per attivare la funzione Lambda

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso ai privilegi limitati per la funzione Lambda
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda
- Abilita il tracciamento di X-Ray
- Impostare le variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws-solutions-buils/aws-events-rule-lambda](https://github.com/aws-solutions-buils/aws-events-rule-lambda)

aws-eventi-rule-sns

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_events_rule_sns</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-events-rule-sns</code> |
|  Java | <code>software.amazon.awsconstructs.services.eventsrulesns</code> |

Overview

Questo modello implementa una regola Amazon CloudWatch Events connessa a un argomento Amazon SNS.

Ecco una definizione minima del modello distribuibile:

```
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);
```

Parametri

- scope [Construct](#)
- id string
- props [EventsRuleToSnsProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|---------------------|----------------------------------|---|
| EventRuleProps | events.RuleProps | Proprietà fornite dall'utente per sovrascrivere le proprietà predefinite per la regola Eventi CloudWatch. |
| Esistente Topicobj? | sns.Topic | Istanza esistente dell'oggetto argomento SNS, fornendo sia questo che topicProps causerà un errore. |
| TopicProps? | sns.TopicProps | Proprietà facoltative fornite dall'utente per ignorare le proprietà predefinite per |

| Nome | Tipo | Descrizione |
|--|---|--|
| | | l'argomento SNS. Ignorato se <code>unexistingTopicObj</code> viene fornito. |
| <code>EnableEncryptionWithCustomerManagedKey?</code> | <code>boolean</code> | Se utilizzare una chiave di crittografia gestita dal cliente, gestita da questa app CDK o importata. Se si importa una chiave di crittografia, questa deve essere specificata nella casella <code>encryptionKey</code> per questo costrutto. |
| <code>encryptionKey?</code> | <code>kms.Key</code> | Chiave di crittografia opzionale esistente da utilizzare al posto della chiave di crittografia predefinita. |
| <code>EncryptionKeyProps?</code> | <code>kms.KeyProps</code> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la chiave di crittografia. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|-------------------------|--|--|
| <code>EventsRule</code> | <code>events.Rule</code> | Restituisce un'istanza della regola Events creata dal pattern. |
| <code>SNSTopic</code> | <code>sns.Topic</code> | Restituisce un'istanza dell'argomento SNS creato dal modello. |

| Nome | Tipo | Descrizione |
|---------------|-------------------------|---|
| encryptionKey | kms.Key | Restituisce un'istanza della chiave di crittografia creata dal modello. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

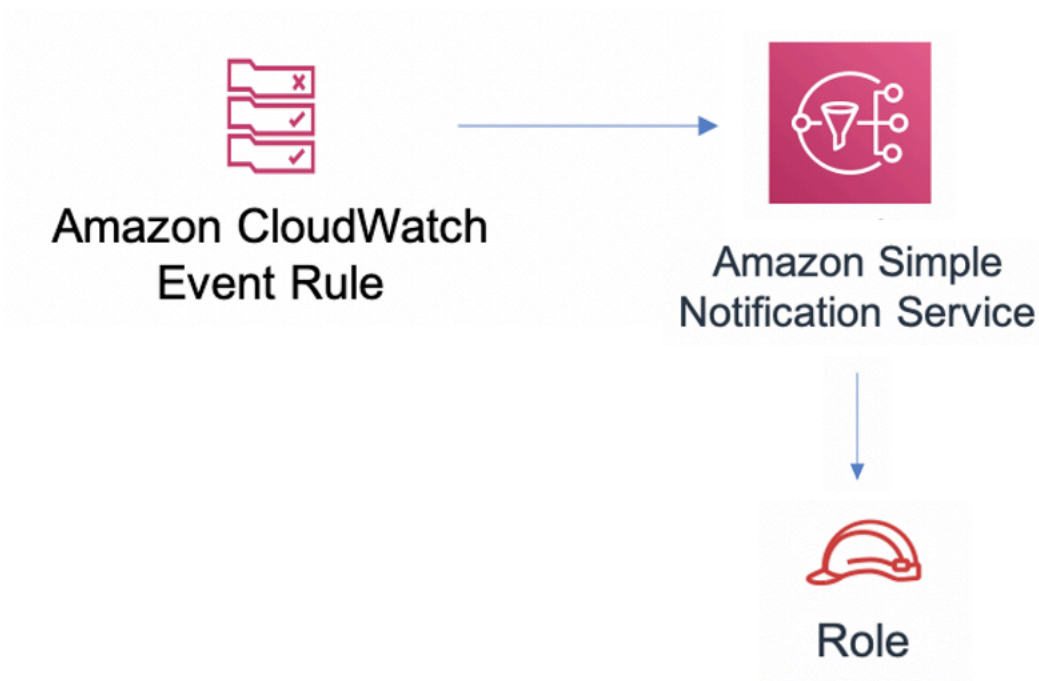
Amazon CloudWatch Events Regola

- Concedere le autorizzazioni di privilegi minimi agli eventi CloudWatch per la pubblicazione nell'argomento SNS.

Argomento Amazon SNS

- Configurare le autorizzazioni di accesso con privilegi minimi per l'argomento SNS.
- Abilitare la crittografia lato server per argomento SNS utilizzando la chiave AWS KMS gestita dal cliente.
- Applicazione della crittografia dei dati in transito

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws-solutions-buils/aws-events-rule-sns](https://github.com/@aws-solutions-buils/aws-events-rule-sns)

aws-eventi-rule-sqs

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario

aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_events_rule_sqs</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-events-rule-sqs</code> |
|  Java | <code>software.amazon.awsconstructs.services.eventsrulesqs</code> |

Overview

Questo modello implementa una regola Amazon CloudWatch Events collegata a una coda Amazon SQS.

Ecco una definizione minima del modello distribuibile:

```
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);
```

Parametri

- scope [Construct](#)
- id [string](#)
- props [EventsRuleToSqsProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|-------------------|----------------------------------|---|
| EventRuleProps | events.RuleProps | Proprietà fornite dall'utente per sovrascrivere le proprietà predefinite per la regola Eventi CloudWatch. |
| ExistingQueueObj? | sqs.Queue | Una coda SQS esistente facoltativa da utilizzare al posto della coda predefinita. Fornire sia questo |

| Nome | Tipo | Descrizione |
|------------------------|---------------------------------------|--|
| | | <code>chequeueProps</code> causerà un errore. |
| QueueProps? | <u>sqs.QueueProps</u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la coda SQS. Ignorato se <code>unexistingQueueObj</code> viene fornito. |
| EnableQueueUrging? | boolean | Indica se concedere autorizzazioni aggiuntive alla funzione Lambda che consente di eliminare la coda SQS. L'impostazione predefinita è <code>false</code> . |
| DeployDeadLetterQueue? | boolean | Se creare una coda secondaria da usare come coda DLQ. L'impostazione predefinita è <code>true</code> . |
| DeadLetterQueueProps? | <u>sqs.QueueProps</u> | Elementi di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per la coda delle lettere morte. Utilizzato solo se il valore <code>deployDeadLetterQueue</code> viene impostata su <code>true</code> . |

| Nome | Tipo | Descrizione |
|---|------------------------------|--|
| MaxReceiveCount? | number | Il numero di volte in cui un messaggio può essere rimosso senza successo prima di essere spostato nella coda DLQ. L'impostazione predefinita è 15. |
| EnableEncryptionWithCustomerManagedKey? | boolean | Se utilizzare una chiave di crittografia gestita dal cliente, gestita da questa app CDK o importata. Se si importa una chiave di crittografia, questa deve essere specificata nella casella <code>encryptionKey</code> per questo costrutto. |
| encryptionKey? | kms.Key | Chiave di crittografia opzionale esistente da utilizzare al posto della chiave di crittografia predefinita. |
| EncryptionKeyProps? | kms.KeyProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la chiave di crittografia. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|------------|-----------------------------|--|
| EventsRule | events.Rule | Restituisce un'istanza della regola Events creata dal pattern. |

| Nome | Tipo | Descrizione |
|------------------|---------------------------|---|
| SQSqueue | sqs.Queue | Restituisce un'istanza della coda SQS creata dal pattern. |
| encryptionKey | kms.Key | Restituisce un'istanza della chiave di crittografia creata dal modello. |
| DeadLetterQueue? | sqs.Queue | Restituisce un'istanza della coda di lettere morte creata dal modello, se ne viene distribuita una. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

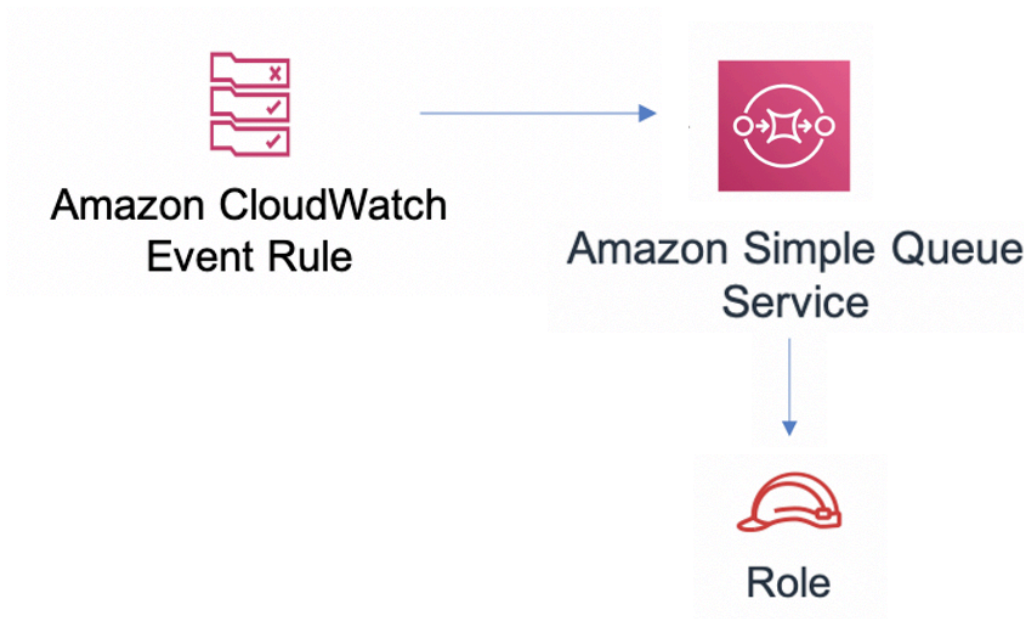
Amazon CloudWatch Events

- Concedere le autorizzazioni di privilegi minimi agli eventi CloudWatch per la pubblicazione nella coda SQS.

Coda Amazon SQS

- Distribuire una coda DLQ per la coda di origine.
- Abilitare la crittografia lato server per la coda di origine utilizzando una chiave AWS KMS gestita dal cliente.
- Applicazione della crittografia dei dati in transito.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws-solutions-builds/aws-event-rule-sqs](https://github.com/aws-solutions-builds/aws-event-rule-sqs)

aws-events-rule-step-function

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic](#)

VersioningModello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_events_rule_step_function</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-events-rule-step-function</code> |
|  Java | <code>software.amazon.awsconstructs.services.eventsrulestepfunction</code> |

Overview

Questo AWS Solutions Construct implementa una regola AWS Events e una funzione AWS Step.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id [string](#)
- props [EventsRuleToStepFunctionProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|------------------------|--|--|
| StateMachineProps | sfn.StateMachinePr ops | Gli oggetti di scena forniti dall'utente facoltativi per sovrascrivere gli oggetti di scena predefiniti per SFN.StateMachine |
| EventRuleProps | events.RuleProps | L'utente ha fornito EventRule Props per sovrascrivere i valori predefiniti |
| CreateCloudWatchalarms | boolean | Indica se creare allarmi CloudWatch consigliati. |
| LogGroupProps? | logs.LogGroupProps | Opzionali degli oggetti di scena forniti dall'utente per sovrascrivere gli oggetti di |

| Nome | Tipo | Descrizione |
|------|------|---|
| | | scena predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------------|---|--|
| CloudWatchArms? | <u>cloudwatch.Alarm[]</u> | Restituisce un elenco di uno o più allarmi CloudWatch creati dal pattern. |
| EventsRule | <u>events.Rule</u> | Restituisce un'istanza della regola Events creata dal pattern. |
| StateMachine | <u>sfn.StateMachine</u> | Restituisce un'istanza della macchina a stati creata dal pattern. |
| StateMachineLogGroup | <u>logs.LogGroup</u> | Restituisce un'istanza del gruppo di log creato dal pattern per la macchina a stati. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

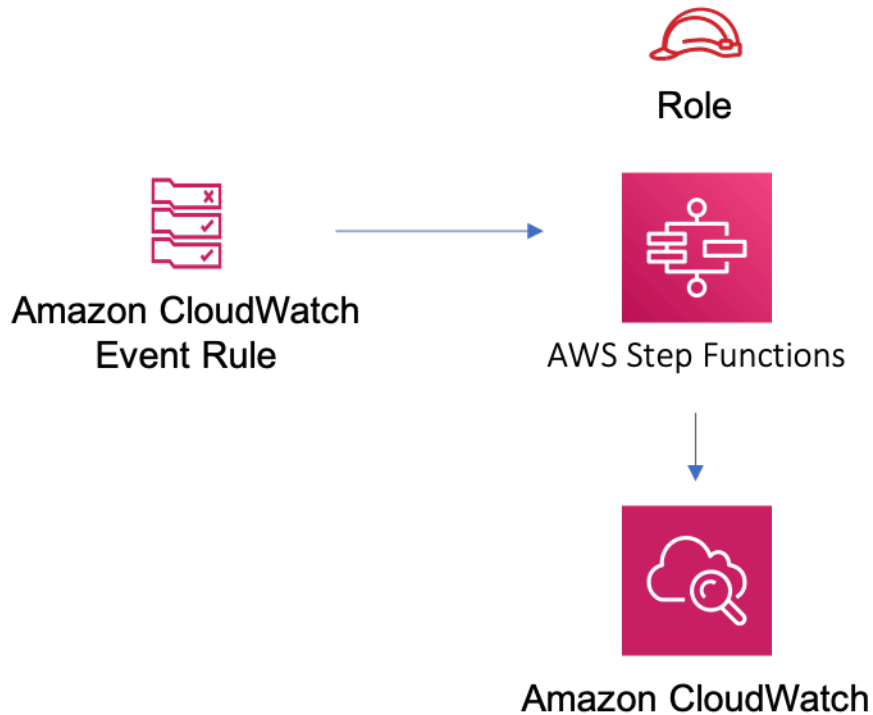
Amazon CloudWatch Events Regola

- Concedere le autorizzazioni di privilegi minimi agli eventi CloudWatch per attivare la funzione Lambda

AWS Step Funzione

- Abilitare la registrazione CloudWatch per API Gateway
- Distribuisci le best practice CloudWatch Allarmi per la funzione Step

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:






[@aws -solutions-constructs/aws-events-rule-step-function](https://github.com/aws-solutions-constructs/aws-events-rule-step-function)

aws-iot-kinesisfirehose-s3

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic Versioning \(ControlloModello\)](#). Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_iot_kinesisfirehose_s3</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-iot-kinesisfirehose-s3</code> |
|  Java | <code>software.amazon.awsconstructs.services.iotkinesisfirehoses3</code> |

Overview

Questo AWS Solutions Construct implementa una regola di argomento AWS IoT MQTT per inviare dati a un flusso di consegna Amazon Kinesis Data Firehose collegato a un bucket Amazon S3.

Ecco una definizione di modello distribuibile minima in 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);

```

Parametri

- scope [Construct](#)
- id [string](#)
- props [IotToKinesisFirehoseToS3Props](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--|--|
| lottoPicruleProps | iot.CfnTopicRulePr ops | L'utente ha fornito CFNTopicRuleProps per sovrascrivere le impostazioni predefinite |
| KinesisFirehosepuns? | kinesisfirehose.Cf nDeliveryStreamPro ps | L'utente opzionale ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per Kinesis Firehose Delivery Stream |
| Esistente BucketObj? | s3.Bucket | Istanza esistente dell'oggetto S3 Bucket, fornendo sia questo che bucketProps In |

| Nome | Tipo | Descrizione |
|----------------|------------------------------------|--|
| | | caso contrario, si verifica un errore |
| Bucket Props? | s3.BucketProps | L'utente ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per il bucket S3. Se questo è fornito, quindi anche <code>fornirebucketProps</code> è un errore. |
| LogGroupProps? | logs.LogGroupProps | Oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|-------------------------|---|--|
| IoTActionsRole | iam.Role | Restituisce un'istanza del ruolo IAM creato dal pattern per la regola IoT. |
| IoTTopicRule | iot.CfnTopicRule | Restituisce un'istanza della regola dell'argomento IoT creata dal pattern. |
| KinesisFireHose | kinesisfirehose.CfnDeliveryStream | Restituisce un'istanza del flusso di consegna Kinesis Firehose creato dal pattern. |
| KinesisFireHoseLogGroup | logs.LogGroup | Restituisce un'istanza del gruppo di log creato dal |

| Nome | Tipo | Descrizione |
|---------------------|---------------------------|---|
| | | modello a cui vengono inviati i log di accesso Kinesis Firehose. |
| KinesisFireHoserole | iam.Role | Restituisce un'istanza del ruolo IAM creato dal pattern per il flusso di consegna Kinesis Firehose. |
| S3Bucket? | s3.Bucket | Restituisce un'istanza del bucket S3 creato dalla serie. |
| S3loggingBucket? | s3.Bucket | Restituisce un'istanza del bucket di registrazione creato dal pattern per il bucket S3. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Regola IoT

- Configurare il ruolo IAM di accesso ai privilegi minimi per Amazon IoT

Amazon Kinesis Firehose

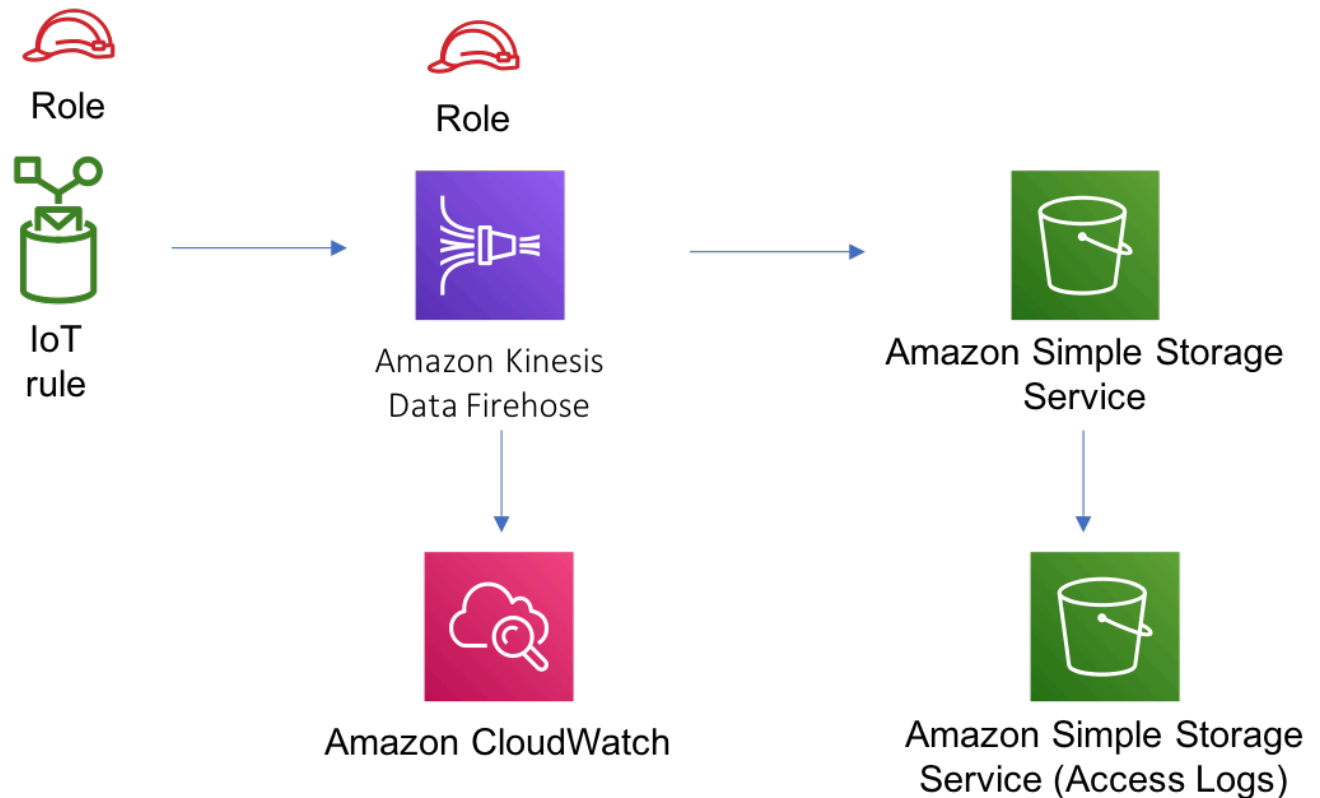
- Abilita la registrazione CloudWatch per Kinesis Firehose
- Configurare il ruolo IAM di accesso ai privilegi minimi per Amazon Kinesis Firehose

Amazon S3

- Configurare la registrazione degli accessi per S3
- Abilitare la crittografia lato server per S3 Bucket utilizzando la chiave AWS gestita
- Attivare il controllo delle versioni per S3 Bucket

- Non consentire l'accesso pubblico per S3 Bucket
- Mantieni il bucket S3 durante l'eliminazione dello stack CloudFormation
- Applica la regola del ciclo di vita per spostare versioni degli oggetti non correnti nell'archivio Glacier dopo 90 giorni

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:






[@aws -solutions-constructs/aws-iot-kinesisfirehose-s3](https://github.com/aws-solutions-constructs/aws-iot-kinesisfirehose-s3)

aws-iot-lambda

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_iot_lambda</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-iot-lambda</code> |
|  Java | <code>software.amazon.awsconstructs.services.iotlambda</code> |

Overview

Questo modello AWS Solutions Constructs implementa una regola di argomento AWS IoT MQTT e un modello di funzione AWS Lambda.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id [string](#)
- props [IotToLambdaProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|---------------------|---------------------------------|--|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFun</code> |

| Nome | Tipo | Descrizione |
|----------------------|--|---|
| | | ctionProps causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se unexistingLambdaObj viene fornito. |
| lottoPicruleProps? | iot.CfnTopicRulePr ops | L'utente ha fornito CFNTopicRuleProps per sovrascrivere le impostazioni predefinite |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------|----------------------------------|--|
| lottoPicrule | iot.CfnTopicRule | Restituisce un'istanza della regola dell'argomento IoT creata dal pattern. |
| LambdaFunction | lambda.Function | Restituisce un'istanza della funzione Lambda creata dal pattern. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

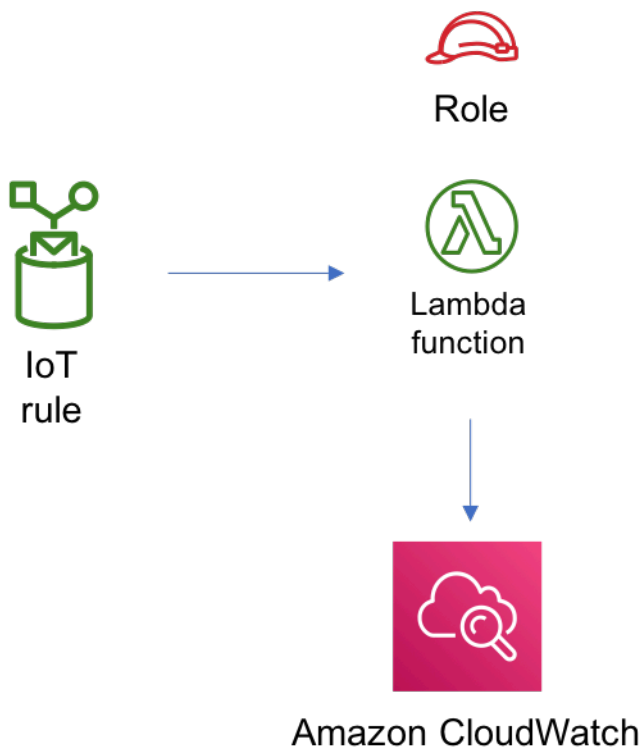
Regola IoT di Amazon

- Configurare il ruolo IAM di accesso con privilegi minimi per Amazon IoT.

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Attivare il tracciamento X-Ray.
- Impostare le variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-buils/aws-iot-lambda](https://github.com/aws-solutions-buils/aws-iot-lambda)


aws-iot-lambda-dynamodb

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic Versioning \(Controllo Modello\)](#). Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_iot_lambda_dynamodb</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-iot-lambda-dynamodb</code> |
|  Java | <code>software.amazon.awsconstructs.services.iotlambdadynamodb</code> |

Overview

Questo modello AWS Solutions Constructs implementa una regola di argomento AWS IoT, una funzione AWS Lambda e una tabella Amazon DynamoDB con le autorizzazioni meno privilegiate.

Ecco una definizione di modello distribuibile minima in 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);

```

Parametri

- `scope` [Construct](#)
- `id` `string`
- `props` [IotToLambdaToDynamoDBProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------------------|---------------------------------|--|
| <code>EsistenteLambdaobj?</code> | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo |

| Nome | Tipo | Descrizione |
|----------------------------------|---|--|
| | | sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| <code>LambdaFunctionProps</code> | <u><code>lambda.FunctionProps</code></u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| <code>IotTopicRuleProps</code> | <u><code>iot.CfnTopicRuleProps</code></u> | Fornito dall'utente di oggetti di scena per sovrascrivere i puntelli predefiniti |
| <code>DynamoTableProps?</code> | <u><code>dynamodb.TableProps</code></u> | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per DynamoDB Table |
| <code>TablePermissions?</code> | <u><code>string</code></u> | Autorizzazioni di tabella facoltative da concedere alla funzione Lambda. È possibile specificare una delle seguenti opzioni: <code>All,Read,ReadWrite</code> , oppure <code>Write</code> . |

Proprietà modelli

| Nome | Tipo | Descrizione |
|--------------------------|--|---|
| <code>DynamoTable</code> | <u><code>dynamodb.Table</code></u> | Restituisce un'istanza della tabella DynamoDB creata dalla serie. |

| Nome | Tipo | Descrizione |
|----------------|---|--|
| lottoPicrule | <u>iot.CfnTopicRule</u> | Restituisce un'istanza della regola dell'argomento IoT creata dal pattern. |
| LambdaFunction | <u>lambda.Function</u> | Restituisce un'istanza della funzione Lambda creata dal pattern. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Regola IoT di Amazon

- Configurare il ruolo IAM di accesso con privilegi minimi per Amazon IoT.

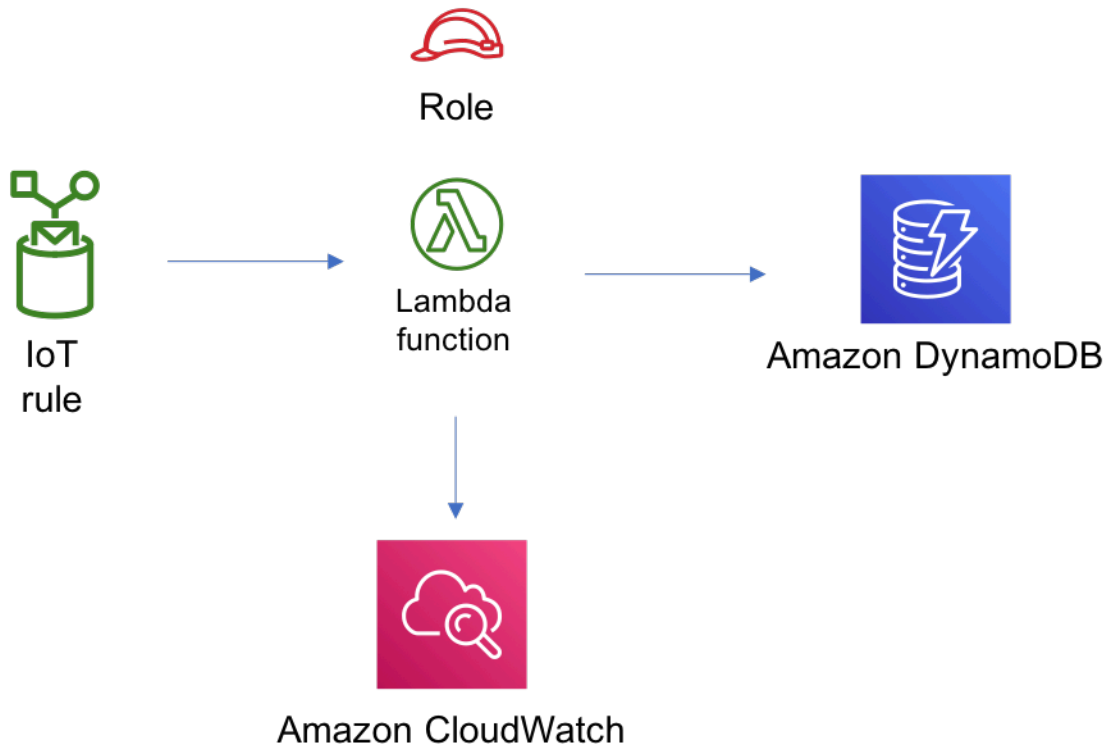
Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Abilitare il tracciamento X-Ray.
- Impostare le variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Tabella Amazon DynamoDB

- Impostare la modalità di fatturazione per DynamoDB Table su richiesta (Paga per richiesta).
- Abilita la crittografia lato server per DynamoDB Table utilizzando la chiave AWS gestita.
- Crea una chiave di partizione chiamata 'id' per DynamoDB Table.
- Mantenere la tabella quando si elimina lo stack CloudFormation.
- Consente i backup continui e il ripristino point-in-time.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-buils/aws-iot-lambda-dyna
modb](https://github.com/aws-solutions-buils/aws-iot-lambda-dynamodb)

aws-kinesisfirehose-s3

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic Versioning](#) ([Controllo](#)Modello). Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws-kinesis-firehose-s3</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-kinesisfirehose-s3</code> |
|  Java | <code>software.amazon.awsconstructs.services.kinesisfirehoses3</code> |

Overview

Questo Construct di soluzioni AWS implementa un flusso di distribuzione di Amazon Kinesis Data Firehose collegato a un bucket Amazon S3.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- `scope` [Construct](#)

- `idstring`
- `props`[KinesisFirehoseToS3Props](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|---------------------------|--|---|
| Bucket Props? | s3.BucketProps | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per il bucket S3. |
| Esistente BucketObj? | s3.IBucket | Istanza esistente opzionale di S3 Bucket. Se questo è fornito, quindi anche <code>fornirebucketProps</code> è un errore. |
| ExistingLoggingBucketObj? | s3.IBucket | Istanza esistente facoltativa di registrazione del bucket S3 per il bucket S3 creato dal pattern. |
| KinesisFirehosepuns? | kinesisfirehose.CfnDeliveryStreamProps any | L'utente opzionale ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per Kinesis Firehose Delivery Stream. |
| LogGroupProps? | logs.LogGroupProps | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per il LogGroup CloudWatchLogs. |

Proprietà pattern

| Nome | Tipo | Descrizione |
|-------------------------|---|---|
| KinesisFireHose | kinesisfirehose.CfnDeliveryStream | Restituisce un'istanza di KinesisFirehose.cfnDeliveryStream creata dal costrutto. |
| KinesisFireHoseLogGroup | logs.LogGroup | Restituisce un'istanza del Logs.LogGroup creato dal costrutto per Kinesis Data Firehose flusso di consegna. |
| KinesisFireHoserole | iam.Role | Restituisce un'istanza del IAM.role creato dal costrutto per il flusso di consegna Kinesis Data Firehose. |
| S3Bucket? | s3.Bucket | Restituisce un'istanza di S3.bucket creata dal costrutto. |
| S3loggingBucket? | s3.Bucket | Restituisce un'istanza di S3.bucket creata dal costrutto come bucket di registrazione per il bucket primario. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

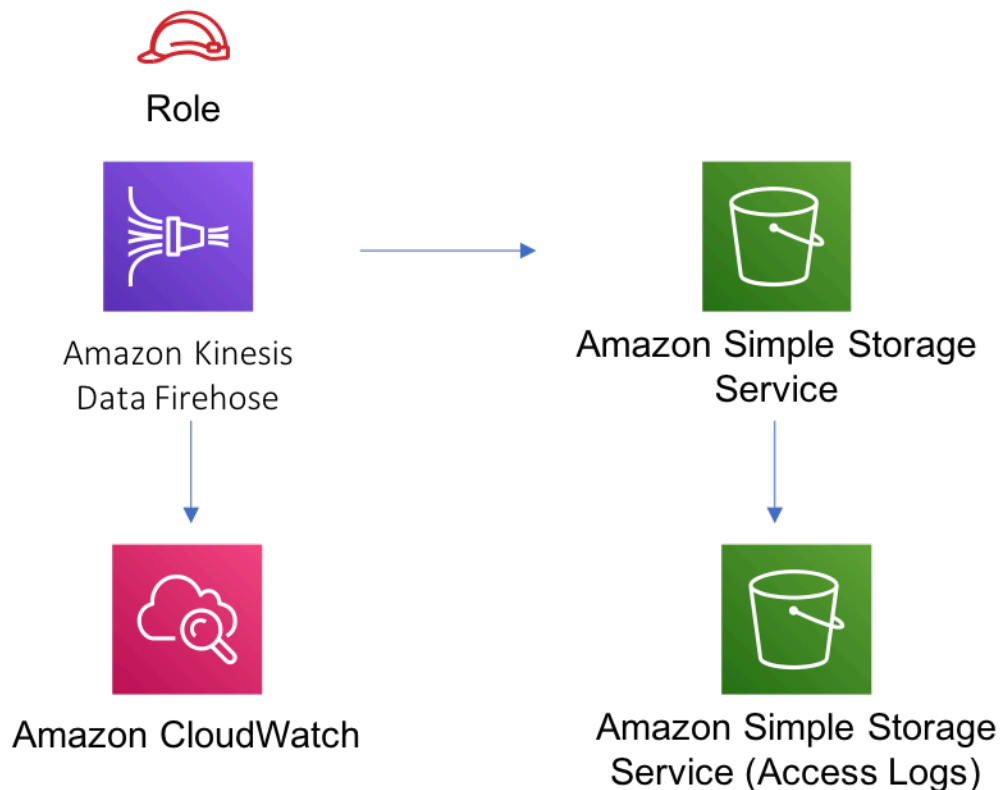
Amazon Kinesis Firehose

- Abilita la registrazione CloudWatch per Kinesis Firehose
- Configurare il ruolo IAM di accesso ai privilegi minimi per Amazon Kinesis Firehose

Amazon S3 Bucket

- Configurare la registrazione degli accessi per S3
- Abilitare la crittografia lato server per S3 Bucket utilizzando la chiave AWS gestita
- Attivare il controllo delle versioni per S3 Bucket
- Non consentire l'accesso pubblico per S3 Bucket
- Mantieni il bucket S3 durante l'eliminazione dello stack CloudFormation
- Applicazione della crittografia dei dati in transito
- Applica la regola del ciclo di vita per spostare versioni degli oggetti non correnti nell'archivio Glacier dopo 90 giorni

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-costruzioni/aws-kinesisfirehose-s3](https://github.com/aws-solutions-costruzioni/aws-kinesisfirehose-s3)




aws-kinesisfirehose-s3-e-kinesisanalytics

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_kinesisfirehose_s3_and_kinesisanalytics</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-kinesisfirehose-s3-and-kinesisanalytics</code> |
|  Java | <code>software.amazon.awsconstructs.services.kinesisfirehose_s3kinesisanalytics</code> |

Overview

Questo AWS Solutions Construct implementa un flusso di consegna Amazon Kinesis Firehose collegato a un bucket Amazon S3 e un'applicazione Amazon Kinesis Analytics.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id `string`
- props [KinesisFirehoseToAnalyticsAndS3Props](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|------------------------|--|---|
| KinesisFirehosepuns? | kinesisFirehose.CfnDeliveryStreamProps | Oggetti di scena opzionali forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il flusso di erogazione Kinesis Firehose. |
| KinesisanalyticsProps? | kinesisAnalytics.CfnApplicationProps | Oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per l'applicazione Kinesis Analytics. |
| Esistente BucketObj? | s3.IBucket | Istanza esistente dell'oggetto S3 Bucket. Se questo è fornito, quindi anche <code>provideBucketProps</code> è un errore. |
| Bucket Props? | s3.BucketProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà |

| Nome | Tipo | Descrizione |
|----------------|---|---|
| LogGroupProps? | <u>logs.LogGroupProps</u> | predefinite per il bucket. Ignorato se unexistingBucketObj viene fornito. Elementi di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|-------------------------|--|--|
| Kinesisanalytics | <u>kinesisAnalytics.CfnApplication</u> | Restituisce un'istanza dell'applicazione Kinesis Analytics creata dal pattern. |
| KinesisFireHose | <u>kinesisfirehose.CfnDeliveryStream</u> | Restituisce un'istanza del flusso di consegna Kinesis Firehose creato dal pattern. |
| KinesisFireHoseLogGroup | <u>logs.LogGroup</u> | Restituisce un'istanza del gruppo di log creato dal modello a cui vengono inviati i log di accesso Kinesis Firehose. |
| KinesisFireHoserole | <u>iam.Role</u> | Restituisce un'istanza del ruolo IAM creato dal pattern per il flusso di consegna Kinesis Firehose. |
| S3Bucket? | <u>s3.Bucket</u> | Restituisce un'istanza del bucket S3 creato dalla serie. |

| Nome | Tipo | Descrizione |
|------------------|---------------------------|---|
| S3loggingBucket? | s3.Bucket | Restituisce un'istanza del bucket di registrazione creato dal pattern per il bucket S3. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Amazon Kinesis Firehose

- Abilita la registrazione CloudWatch per Kinesis Firehose
- Configurare il ruolo IAM di accesso ai privilegi minimi per Amazon Kinesis Firehose

Amazon S3

- Configurare la registrazione degli accessi per S3
- Abilita la crittografia lato server per S3 Bucket utilizzando la chiave AWS gestita
- Attivare il controllo delle versioni per S3 Bucket
- Non consentire l'accesso pubblico per S3 Bucket
- Mantenere il bucket S3 quando si elimina lo stack CloudFormation
- Applicazione della crittografia dei dati in transito
- Applica la regola del ciclo di vita per spostare versioni degli oggetti non correnti nell'archivio Glacier dopo 90 giorni

Amazon Kinesis Data Analytics

- Configurare il ruolo IAM di accesso ai privilegi minimi per Amazon Kinesis Analytics

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:






[@aws -solutions-constructs/aws-kinesisfirehose-s3-and-kinesisanalytics](#)

aws-kinesisstreams-colla

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti all'[controllo semantico delle versioni](#)Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_kinesis_streams_gluejob</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-kinesisstreams-gluejob</code> |
|  Java | <code>software.amazon.awsconstructs.services.kinesisstreamsgluejob</code> |

Overview

Questo Construct di soluzioni AWS distribuisce un flusso di dati Amazon Kinesis e configura un AWS Glue Job per eseguire la trasformazione ETL personalizzata con le risorse/proprietà appropriate per l'interazione e la sicurezza. Crea anche un bucket Amazon S3 in cui è possibile caricare lo script Python per AWS Glue Job.

Ecco una definizione di modello distribuibile minima in 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);

```

Parametri

- `scope` [Construct](#)
- `id` `string`
- `props` [KinesisstreamsToGluejobProps](#)

Puntelli di costruzione del modello

| Nome | Tipo | Descrizione |
|------------------------|--|--|
| KinesisStreamProps? | <u>kinesis.StreamProps</u> | Accessori di scena facoltativi forniti dall'utente per ignorare gli oggetti di scena predefiniti per il flusso di dati Amazon Kinesis. |
| Streamobj esistenteBJ? | <u>kinesis.Stream</u> | Istanza esistente di Kinesis Stream, fornendo sia questo che <code>kinesisStreamProps</code> causerà un errore. |
| GluejobProps? | <u>cfnJob.CfnJobProps</u> | Gli oggetti di scena forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il processo AWS Glue. |
| EsistingGluejob? | <u>cfnJob.CfnJob</u> | Istanza esistente di AWS Glue Job, fornendo sia questo che <code>glueJobProps</code> causerà un errore. |
| Database esistente? | <u>CfnDatabase</u> | Database AWS Glue esistente da utilizzare con questo costruito. Se questo è impostato, allora <code>databaseProps</code> Viene ignorato. |
| DatabaseProps? | <u>CfnDatabaseProps</u> | Gli oggetti di scena forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti utilizzati per creare il database AWS Glue. |

| Nome | Tipo | Descrizione |
|--------------------|---|--|
| Tabella esistente? | CfnTable | Istanza esistente della tabella AWS Glue. Se questo è impostato, allora <code>tableProps</code> e <code>fieldSchema</code> Vengono ignorati. |
| TableProps? | CfnTableProps | Gli oggetti di scena forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti utilizzati per creare una tabella AWS Glue. |
| Schema di campo? | CfnTable.ColumnProperty[] | Struttura dello schema fornita dall'utente per creare una tabella AWS Glue. |
| OutputDataStore? | SinkDataStoreProps | Puntelli forniti dall'utente per un bucket Amazon S3 che memorizza l'output dal lavoro AWS Glue. Attualmente supporta solo Amazon S3 come tipo di datastore di output. |

SinkDataStoreProps

| Nome | Tipo | Descrizione |
|-------------------------|------------------------|---|
| ExistingS3OutputBucket? | Bucket | Istanza esistente del bucket S3 in cui i dati devono essere scritti. Fornire sia questo che <code>outputBucketProps</code> causerà un errore. |

| Nome | Tipo | Descrizione |
|-------------------|-------------------------------|--|
| OutputBucketProps | BucketProps | Proprietà bucket fornite dall'utente per creare il bucket Amazon S3 utilizzato per memorizzare l'output dal processo AWS Glue. |
| DataStoreType | SinkStoreType | Tipo di archivio dati sink. |

SinkStoreType

Enumerazione dei tipi di Data Store che possono includere S3, DynamoDB, DocumentDB, RDS o Redshift. L'implementazione corrente del costrutto supporta solo S3, ma potrebbe aggiungere altri tipi di output in futuro.

| Nome | Tipo | Descrizione |
|------|--------|--------------------|
| S3 | string | Tipo di storage S3 |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Amazon Kinesis Streams

- Configurare il ruolo IAM di accesso con privilegi minimi per il flusso di dati Amazon Kinesis.
- Abilita la crittografia lato server per Amazon Kinesis Stream utilizzando una chiave AWS Managed KMS.
- Distribuisci le best practice di Amazon CloudWatch Allarmi per Amazon Kinesis Stream.

Glue Job

- Crea una configurazione di sicurezza AWS Glue che configura la crittografia per CloudWatch, Job Bookmarks e S3. CloudWatch e Job Bookmarks sono crittografati utilizzando AWS Managed KMS Key creata per AWS Glue Service. Il bucket S3 è configurato con la modalità di crittografia SSE-S3.
- Configurare i criteri di ruolo del servizio che consentono a AWS Glue di leggere da Amazon Kinesis Data Streams.

Glue Database

- Creare un database AWS Glue. Una tabella AWS Glue verrà aggiunta al database. Questa tabella definisce lo schema per i record memorizzati nel flusso di dati Amazon Kinesis.

Glue Tabella

- Creare una tabella AWS Glue. La definizione dello schema della tabella si basa sulla struttura JSON dei record memorizzati nel flusso di dati Amazon Kinesis.

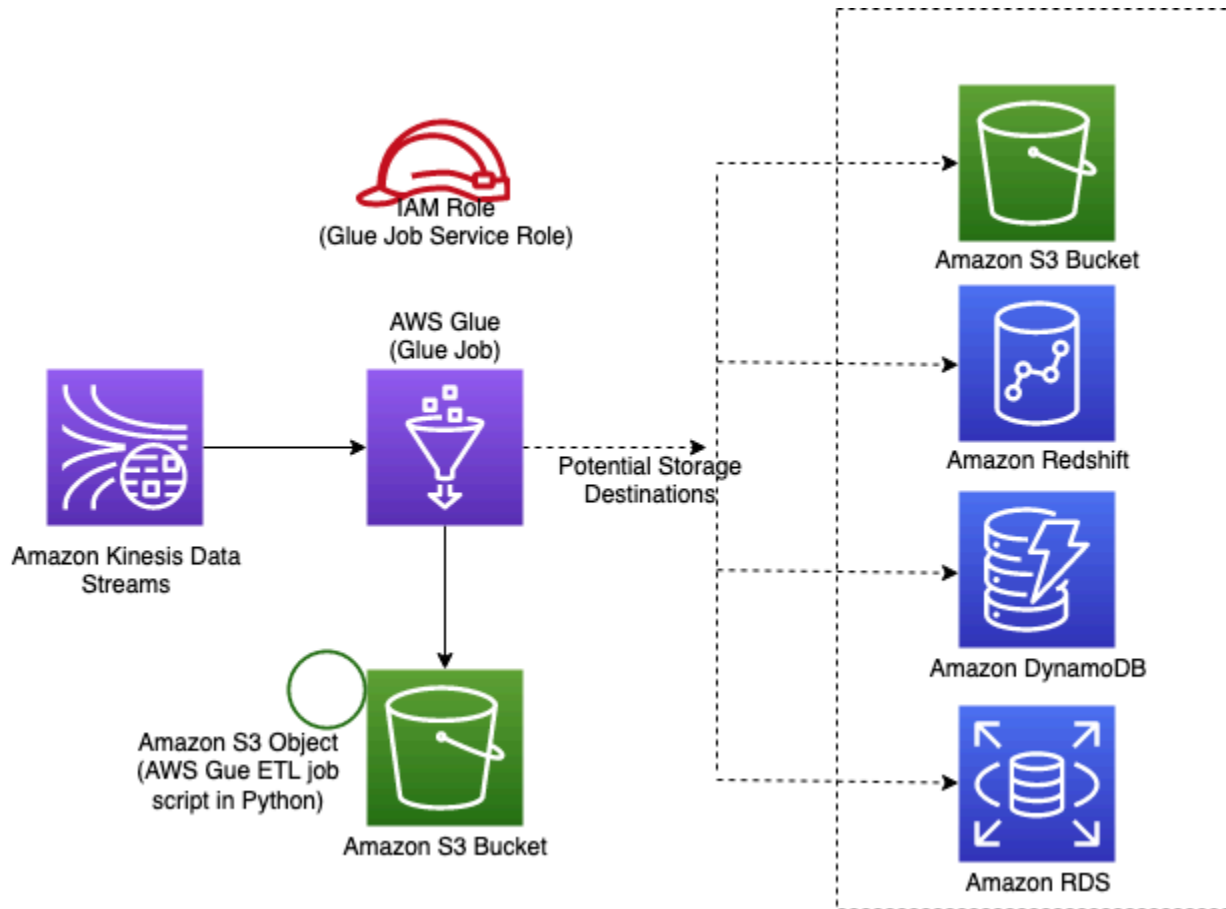
Ruolo IAM

- Ruolo di esecuzione del processo che dispone dei privilegi per 1) leggere lo script ETL dalla posizione del bucket Amazon S3, 2) leggere i record dal flusso di dati Amazon Kinesis e 3) eseguire il processo Amazon Glue.

Perlue di output

- Un bucket Amazon S3 in cui archiviare l'output della trasformazione ETL. Questo bucket verrà passato come argomento al lavoro AWS Glue creato in modo che possa essere utilizzato nello script ETL per scrivere dati in esso.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws-solutions-constructs/aws-kinesisstreams-gluejob](https://github.com/aws-solutions-constructs/aws-kinesisstreams-gluejob)




aws-kinesisstreams-kinesisfirehose-s3

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic versioning](#) ([Controllo delle](#)Modello). Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario

aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | aws_solutions_constructs.aws_kinesisstreams_kinesisfirehose_s3 |
|  TypeScript | @aws-solutions-constructs/aws-kinesis-streams-kinesis-firehose-s3 |
|  Java | software.amazon.awsconstructs.services.kinesisstreamskinesisfirehoses3 |

Overview

Questo Construct di soluzioni AWS implementa un flusso di dati Kinesis (KDS) di Amazon collegato a Amazon Kinesis Data Firehose (KDF) flusso di consegna collegato a un bucket Amazon S3.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- idstring
- props [KinesisStreams...ToS3Props](#)

Schema di costruzione

| Nome | Tipo | Descrizione |
|---------------------------|--------------------------------|---|
| Bucket Props? | s3.BucketProps | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per il bucket S3. |
| CreateCloudWatchalarms? | boolean | Facoltativo se creare allarmi CloudWatch consigliati. |
| Esistente BucketObj? | s3.IBucket | Istanza esistente facoltativa dell'oggetto S3 Bucket. Se questo è fornito, quindi anche fornire bucketProps è un errore. |
| ExistingLoggingBucketObj? | s3.IBucket | Istanza esistente facoltativa di registrazione dell'oggetto Bucket S3 per il Bucket S3 creato dal pattern. |
| StreamObj esistente? | kinesis.Stream | Istanza esistente di Kinesis Stream, fornendo sia questo che <code>kinesisStreamProps</code> . In caso contrario, si verifica un errore. |

| Nome | Tipo | Descrizione |
|-----------------------|--|--|
| KinesisFirehosepumps? | aws-kinesisfirehose.CfnDeliveryStreamProps any | L'utente opzionale ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per Kinesis Firehose Delivery Stream. |
| KinesisStreamProps? | kinesis.StreamProps | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per il flusso Kinesis. |
| LogGroupProps? | logs.LogGroupProps | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatchLogs. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|-------------------------|---|---|
| CloudwatchArms? | cloudwatch.Alarm[] | Restituisce un elenco di istanze CloudWatch.Alarm create dal costruito. |
| KinesisFireHose | kinesisfirehose.CfnDeliveryStream | Restituisce un'istanza di KinesisFirehose.cfnDeliveryStream creata dal costruito. |
| KinesisFireHoseLogGroup | logs.LogGroup | Restituisce un'istanza del Logs.LogGroup creato dal costruito per Kinesis Data Firehose flusso di consegna. |

| Nome | Tipo | Descrizione |
|---------------------|---------------------------|---|
| KinesisFireHoseRole | iam.Role | Restituisce un'istanza del IAM.role creato dal costrutto per il flusso di consegna Kinesis Data Firehose. |
| KinesisStreamRole | iam.Role | Restituisce un'istanza del IAM.role creato dal costrutto per il flusso Kinesis. |
| S3Bucket? | s3.Bucket | Restituisce un'istanza di S3.bucket creata dal costrutto. |
| S3loggingBucket? | s3.Bucket | Restituisce un'istanza di S3.bucket creata dal costrutto come bucket di registrazione per il bucket primario. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Amazon Kinesis Streaming

- Configurare il ruolo IAM di accesso ai privilegi minimi per Kinesis Stream
- Abilitare la crittografia lato server per Kinesis Stream utilizzando AWS Managed KMS Key
- Distribuisci le best practice CloudWatch Allarmi per Kinesis Stream

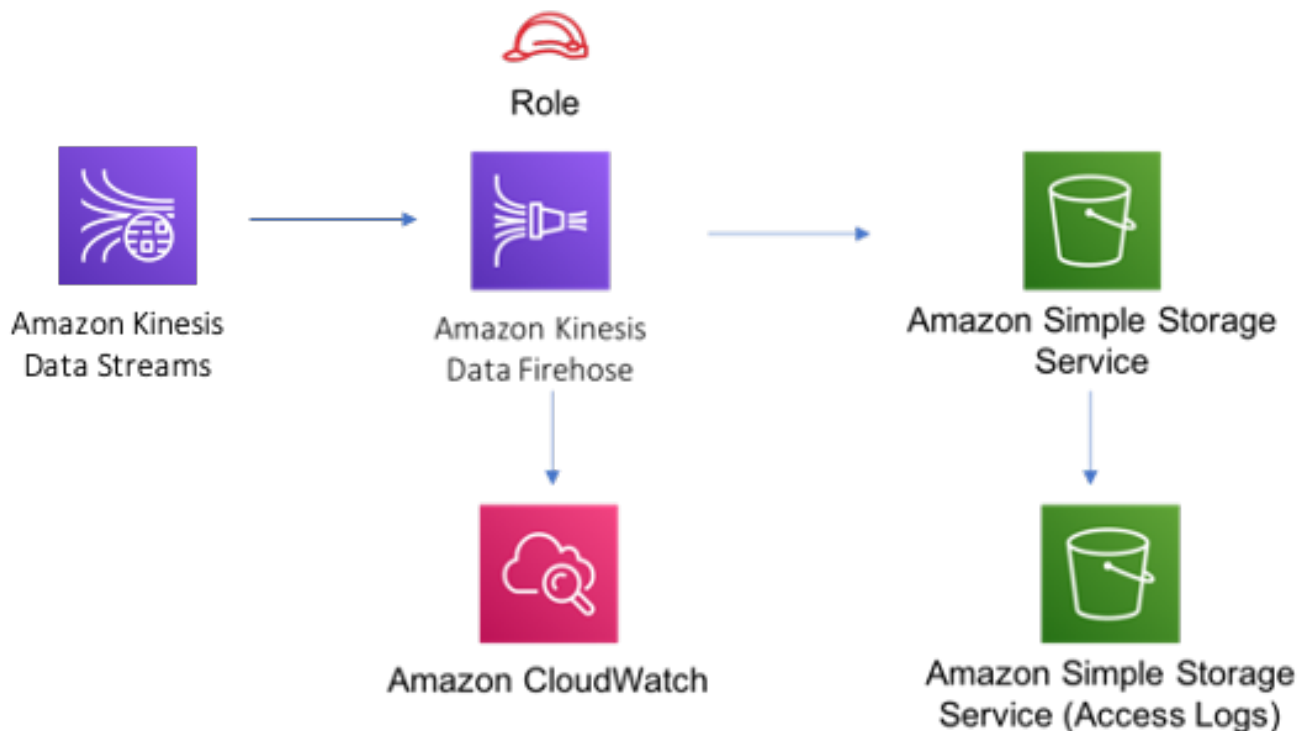
Amazon Kinesis Firehose

- Abilita la registrazione CloudWatch per Kinesis Firehose
- Configurare il ruolo IAM di accesso ai privilegi minimi per Amazon Kinesis Firehose

Amazon S3

- Configurare la registrazione degli accessi per il bucket
- Abilitare la crittografia lato server per il bucket S3 utilizzando la chiave AWS gestita KMS
- Applicazione della crittografia dei dati in transito
- Abilita il controllo delle versioni del bucket
- Non consentire l'accesso pubblico per il bucket S3
- Mantenere il bucket S3 quando si elimina lo stack CloudFormation
- Applica la regola del ciclo di vita per spostare versioni degli oggetti non correnti nell'archivio Glacier dopo 90 giorni

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:




[@aws -solutions-costruzioni/aws-kinesisstreams-kinesisfirehose-s3](https://github.com/@aws-solutions-costruzioni/aws-kinesisstreams-kinesisfirehose-s3)

aws-kinesisstreams-lambda

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Controllo delle versioni semantiche](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|--|--|
|  Python | <code>aws_solutions_constructs.aws-kinesis-streams-lambda</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-kinesisstreams-lambda</code> |
|  Java | <code>software.amazon.awsconstructs.services.kinesisstreamslambda</code> |

Overview

Questo Construct di soluzioni AWS distribuisce una funzione Kinesis Stream e Lambda con le risorse/proprietà appropriate per l'interazione e la sicurezza.

Ecco una definizione di modello distribuibile minima in 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);

```

Parametri

- scope [Construct](#)
- id [string](#)
- props [KinesisStreamsToLambdaProps](#)

Parametri di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--------------------------------------|---|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite |

| Nome | Tipo | Descrizione |
|--------------------------|---|---|
| KinesisStreamProps? | <u>kinesis.StreamProps</u> | te per la funzione Lambda. Ignorato se unexistingL ambdaObj viene fornito. Oggetti di scena opzionali forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il flusso Kinesis. |
| Streamobj esistenteBJ? | <u>kinesis.Stream</u> | Istanza esistente di Kinesis Stream, fornendo sia questo chekinesisStreamProps causerà un errore. |
| KinesiSeventSourceProps? | <u>aws-lambda-event-sources.KinesisEventSourceProps</u> | Oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per la mappatura dell'origine degli eventi Lambda. |
| CreateCloudWatchalarms | boolean | Indica se creare allarmi CloudWatch consigliati. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------|--|--|
| KinesisStream | <u>kinesis.Stream</u> | Restituisce un'istanza del flusso Kinesis creato dal pattern. |
| LambdaFunction | <u>lambda.Function</u> | Restituisce un'istanza della funzione Lambda creata dal pattern. |

| Nome | Tipo | Descrizione |
|-------------------|------------------------------------|--|
| KinesisStreamRole | iam.Role | Restituisce un'istanza del ruolo IAM creato dal pattern per il flusso Kinesis. |
| CloudwatchArms? | cloudwatch.Alarm[] | Restituisce un elenco di uno o più avvisi CloudWatch creati dal pattern. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

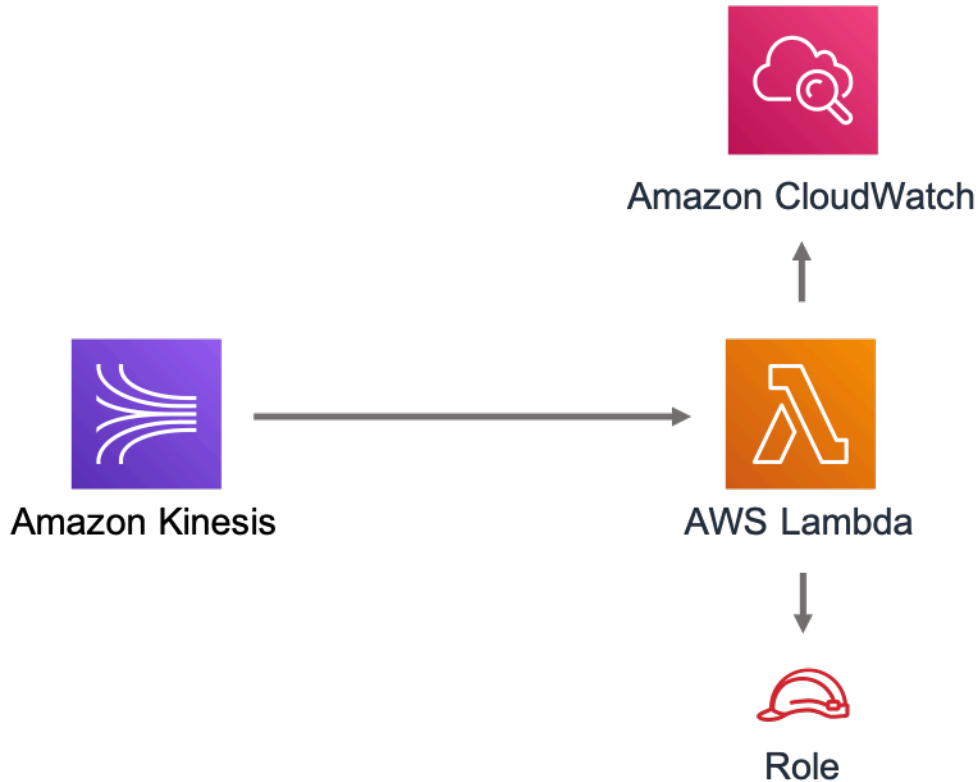
Amazon Kinesis Stream

- Configurare il ruolo IAM di accesso ai privilegi minimi per Kinesis Stream.
- Abilita la crittografia lato server per Kinesis Stream utilizzando la chiave di gestione delle chiavi gestita AWS.
- Distribuisci le best practice CloudWatch Allarmi per Kinesis Stream.

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Attivare il tracciamento X-Ray.
- Abilita funzionalità di gestione degli errori: abilita la bisezione sulla funzione Errore; imposta la durata massima record predefinita (24 ore); imposta i tentativi di tentativi massimi predefiniti (500) e distribuisci la coda di lettere non recapitate SQS come destinazione in caso di errore.
- Impostare le variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:





[@aws -solutions-costruzioni/aws-kinesisstreams-lambda](#)

aws-lambda-dynamodb

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti all'[controllo delle versioni semantiche](#)Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_lambda_dynamodb</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-lambda-dynamodb</code> |
|  Java | <code>software.amazon.awsconstructs.services.lambda.dynamodb</code> |

Overview

Questo AWS Solutions Construct implementa la funzione AWS Lambda e la tabella Amazon DynamoDB con autorizzazioni con privilegi minimi.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id `string`
- props [LambdaToDynamoDBProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--------------------------------------|--|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| DynamoTableProps? | dynamodb.TableProps | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per DynamoDB Table |
| TableObj esistente? | dynamodb.Table | Istanza esistente dell'oggetto tabella DynamoDB, fornendo |

| Nome | Tipo | Descrizione |
|-------------------------------|---------------------------------------|---|
| | | sia questo che <code>dynamoTableProps</code> causerà un errore. |
| TablePermissions? | <code>string</code> | Autorizzazioni di tabella facoltative da concedere alla funzione Lambda. Può essere specificata una delle seguenti opzioni: <code>All,Read,ReadWrite</code> , oppure <code>Write</code> . |
| TableEnvironmentVariableName? | <code>string</code> | Nome facoltativo per la variabile di ambiente della tabella DynamoDB impostata per la funzione Lambda. |
| VPC esistente? | <code>ec2.IVpc</code> | Un VPC opzionale esistente in cui deve essere distribuito questo modello. Quando viene distribuita in un VPC, la funzione Lambda utilizzerà ENI nel VPC per accedere alle risorse di rete e verrà creato un endpoint gateway nel VPC per Amazon DynamoDB. Se viene fornito un VPC esistente, <code>deployVpc</code> non può essere <code>true</code> . Questo utilizza <code>ec2.IVpc</code> per consentire ai client di fornire VPC che esistono al di fuori dello stack utilizzando il metodo <code>ec2.Vpc.fromLookup()</code> Metodo. |

| Nome | Tipo | Descrizione |
|------------|------------------------------|--|
| VPCProp? | ec2.VpcProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per il nuovo VPC. <code>enableDns Hostnames</code> , <code>enableDns Support</code> , <code>natGateways</code> , <code>subnetConfiguratio n</code> sono impostati dal modello, quindi tutti i valori per quelle proprietà fornite qui verranno ignorati. <code>SedeployVpc</code> non è <code>true</code> allora questa proprietà verrà ignorata. |
| DeployVPC? | boolean | <p>Se creare un nuovo VPC basato su <code>vpcProps</code> in cui distribuire questo modello. Impostando questo su <code>true</code> verrà distribuito il VPC minimo e più privato per eseguire il pattern:</p> <ul style="list-style-type: none"> • Una sottorete isolata in ogni zona di disponibilità utilizzata dal programma CDK • <code>enableDnsHostnames</code> e <code>enableDns Support</code> saranno entrambi impostati su <code>true</code> <p>Se questa proprietà è <code>true</code>, quindi <code>existingVpc</code> non può essere specificato. L'impostazione predefinita è <code>false</code>.</p> |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------|---------------------------------|--|
| DynamoTable | dynamodb.Table | Restituisce un'istanza della tabella DynamoDB creata dalla serie. |
| LambdaFunction | lambda.Function | Restituisce un'istanza della funzione Lambda creata dal pattern. |
| VPC? | ec2.IVpc | Restituisce un'interfaccia sul VPC utilizzato dal modello (se presente). Questo può essere un VPC creato dal pattern o dal VPC fornito al costruttore del pattern. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Funzione di AWS Lambda

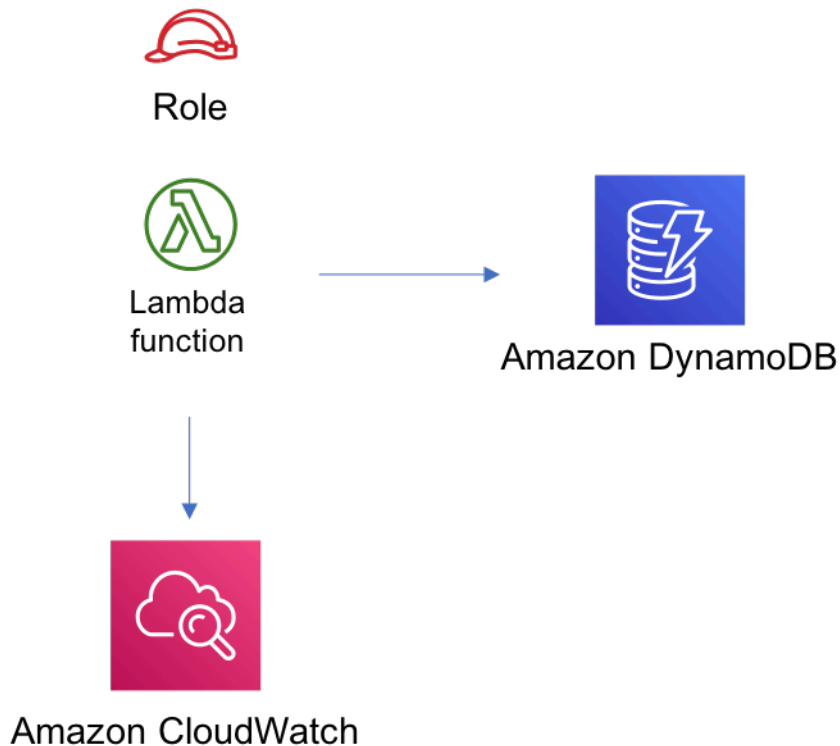
- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Attivare il tracciamento X-Ray.
- Impostazione delle variabili di ambiente:
 - DDB_TABLE_NAME (predefinito)
 - AWS_NODEJS_CONNECTION_REUSE_ENABLED(per le funzioni Node 10.x e successive)

Tabella Amazon DynamoDB

- Impostare la modalità di fatturazione per DynamoDB Table su richiesta (Paga per richiesta).

- Abilitare la crittografia lato server per DynamoDB Table utilizzando la chiave di gestione delle chiavi gestita AWS.
- Crea una chiave di partizione chiamata 'id' per DynamoDB Table.
- Mantenere la tabella quando si elimina lo stack CloudFormation.
- Backup continui e ripristino point-in-time.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-constructs/aws-lambda-dynamodb](https://github.com/aws-solutions-constructs/aws-lambda-dynamodb)




aws-lambda-elasticsearch-kibana

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic Versioning \(Controllo Modello\)](#). Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_lambda_elasticsearch_kibana</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-lambda-elasticsearch-kibana</code> |
|  Java | <code>software.amazon.awsconstructs.services.lambdaelasticsearchkibana</code> |

Overview

Questo AWS Solutions Construct implementa una funzione AWS Lambda e un dominio Amazon Elasticsearch Service con autorizzazioni meno privilegiate.

Ecco una definizione di modello distribuibile minima in 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);

```

Parametri

- scope [Construct](#)
- id [string](#)
- props [LambdaToElasticSearchAndKibanaProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|---------------------|---------------------------------|---|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |

| Nome | Tipo | Descrizione |
|--|---|---|
| LambdafunctionPunps? | <u>lambda.FunctionProps</u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se unexistingLambdaObj viene fornito. |
| EsDomainProps? | <u>elasticsearch.CfnDomainProps</u> | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per il Amazon Elasticsearch Service |
| domainName | string | Nome di dominio per il servizio Cognito e Amazon Elasticsearch |
| CognitoDomainName? | string | Nome di dominio Cognito facoltativo. Se fornito, verrà utilizzato per il dominio Cognito edomainName verrà utilizzato per il dominio Elasticsearch. |
| CreateCloudWatchalarms | boolean | Indica se creare allarmi CloudWatch consigliati. |
| DomainEndPointEnvironmentVariableName? | string | Nome facoltativo per la variabile di ambiente endpoint dominio ElasticSearch impostata per la funzione Lambda. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|-------------------------|---|---|
| CloudwatChAlarms? | cloudwatch.Alarm[] | Restituisce un elenco di uno o più avvisi CloudWatch creati dal pattern. |
| ElasticSearchDomain | elasticsearch.CfnDomain | Restituisce un'istanza del dominio Elasticsearch creato dal pattern. |
| ElasticSearchDomainRole | iam.Role | Restituisce un'istanza del ruolo IAM creato dal pattern per il dominio Elasticsearch. |
| IdentityPool | cognito.CfnIdentityPool | Restituisce un'istanza del pool di identità Cognito creato dal pattern. |
| LambdaFunction | lambda.Function | Restituisce un'istanza della funzione Lambda creata dal pattern. |
| userPool | cognito.UserPool | Restituisce un'istanza del pool di utenti Cognito creato dal pattern. |
| userPoolClient | cognito.UserPoolClient | Restituisce un'istanza del client del pool di utenti Cognito creato dal pattern. |

valida e completa

Questo modello richiede una funzione Lambda in grado di pubblicare dati nel servizio Elasticsearch dal flusso DynamoDB. Viene fornita una funzione di esempio [Qui](#): .

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Abilita il tracciamento X-Ray.
- Impostare le variabili di ambiente:
 - DOMAIN_ENDPOINT (predefinito)
 - AWS_NODEJS_CONNECTION_REUSE_ENABLED(per le funzioni Node 10.x e successive)

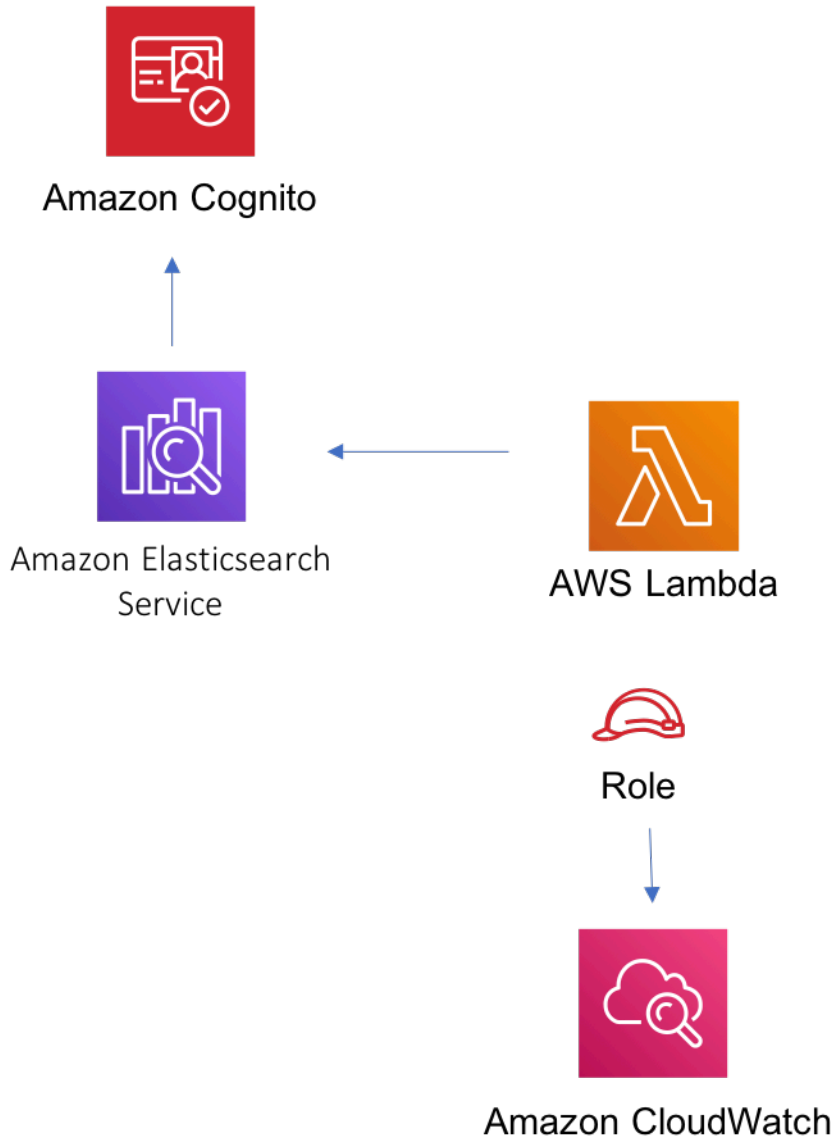
Amazon Cognito

- Impostare i criteri password per i pool di utenti.
- Applicare la modalità di protezione avanzata per i pool di utenti.

Amazon Elasticsearch Service

- Distribuisci le best practice CloudWatch Allarmi per il dominio Elasticsearch.
- Proteggi l'accesso alla dashboard di Kibana con i pool di utenti di Cognito.
- Abilita la crittografia lato server per il dominio Elasticsearch utilizzando la chiave di gestione delle chiavi gestita AWS.
- Abilita la crittografia da nodo a nodo per il dominio Elasticsearch.
- Configurare il cluster per il dominio Amazon ES.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:






[@aws -solutions-constructs/aws-lambda-elasticsearch-kibana](https://github.com/aws-solutions-constructs/aws-lambda-elasticsearch-kibana)

aws-lambda-s3

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Controllo delle versioni semantiche](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|--|--|
|  Python | <code>aws_solutions_constructs.aws_lambda_s3</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-lambda-s3</code> |
|  Java | <code>software.amazon.awsconstructs.services.lambdas3</code> |

Overview

Questo Construct di soluzioni AWS implementa una funzione Lambda collegata a un bucket Amazon S3.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id [string](#)
- props [LambdaToS3Props](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--------------------------------------|--|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| Esistente BucketObj? | s3.IBucket | Istanza esistente dell'oggetto S3 Bucket. Se questo |

| Nome | Tipo | Descrizione |
|--------------------|--------------------------------|---|
| | | è fornito, quindi anche fornirebucketProps è un errore. |
| Bucket Props? | s3.BucketProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per il bucket. Ignorato se unexistingBucketObj viene fornito. |
| BucketPermissions? | string[] | Autorizzazioni bucket facoltative da concedere alla funzione Lambda. Può essere specificato uno o più dei valori seguenti: Delete,Put,Read,ReadWrite,Write: . |

| Nome | Tipo | Descrizione |
|----------------|--------------------------|---|
| VPC esistente? | ec2.IVpc | <p>Un VPC opzionale esistente in cui deve essere distribuito questo modello. Quando viene distribuita in un VPC, la funzione Lambda utilizzerà ENI nel VPC per accedere alle risorse di rete e verrà creato un endpoint di interfaccia nel VPC per Amazon SQS. Se viene fornito un VPC esistente, <code>deployVpc</code> non può essere <code>true</code>. Questo utilizza <code>ec2.IVpc</code> per consentire ai client di fornire i VPC che esistono al di fuori dello stack utilizzando il ec2.Vpc.fromLookup() Metodo.</p> |

| Nome | Tipo | Descrizione |
|------------|---------|---|
| DeployVPC? | boolean | <p>Se creare un nuovo VPC basato su <code>vpcProps</code> in cui distribuire questo modello. Impostazione di questo valore <code>true</code> distribuirà il VPC minimo e più privato per eseguire il pattern:</p> <ul style="list-style-type: none">• Una sottorete isolata in ogni zona di disponibilità utilizzata dal programma CDK.• <code>enableDnsHostnames</code> e <code>enableDnsSupport</code> saranno entrambi impostati su <code>true</code>. <p>Se questa proprietà è <code>true</code>, quindi <code>existingVpc</code> non può essere specificato. L'impostazione predefinita è <code>false</code>.</p> |

| Nome | Tipo | Descrizione |
|--------------------------------|------------------------------|---|
| VPCProp? | ec2.VpcProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per il nuovo VPC. <code>enableDnsHostnames</code> , <code>enableDnsSupport</code> , <code>natGatewaySubnetConfiguration</code> sono impostati dal modello, quindi tutti i valori per quelle proprietà fornite qui verranno ignorati. <code>SeedployVpc</code> non è <code>true</code> . Questa proprietà verrà ignorata. |
| bucketEnvironmentVariableName? | string | Nome facoltativo per la variabile di ambiente bucket S3 impostata per la funzione Lambda. |

Proprietà dei modelli

| Nome | Tipo | Descrizione |
|------------------|---------------------------------|---|
| LambdaFunction | lambda.Function | Restituisce un'istanza della funzione Lambda creata dal pattern. |
| S3Bucket? | s3.Bucket | Restituisce un'istanza del bucket S3 creato dalla serie. |
| S3loggingBucket? | s3.Bucket | Restituisce un'istanza del bucket di registrazione creato dal pattern per il bucket S3. |

| Nome | Tipo | Descrizione |
|---------|--------------------------|--|
| Il VPC? | ec2.IVpc | Restituisce un'istanza del VPC utilizzato dal modello (se presente). Questo può essere un VPC creato dal pattern o dal VPC fornito al costruttore del pattern. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Abilita traccia X-Ray
- Imposta le variabili di ambiente:
 - S3_BUCKET_NAME (predefinito)
 - AWS_NODEJS_CONNECTION_REUSE_ENABLED(per le funzioni Node 10.x e successive)

S3 Amazon S3

- Configurare la registrazione accesso per S3 Bucket
- Abilita la crittografia lato server per S3 Bucket utilizzando la chiave AWS gestita KMS.
- Attivare il controllo delle versioni per S3 Bucket.
- Non consentire l'accesso pubblico per S3 Bucket.
- Mantenere il bucket S3 quando si elimina lo stack CloudFormation.
- Applicazione della crittografia dei dati in transito
- Applica la regola del ciclo di vita per spostare le versioni degli oggetti non correnti nell'archivio Glacier dopo 90 giorni.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:






[@aws-solutions-constructs/aws-lambda-s3](https://github.com/aws-solutions-constructs/aws-lambda-s3)

aws-lambda-ssmstringparameter

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti all'[controllo delle versioni semantiche](#)Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_contracts.aws_lambda_ssm_string_parameter</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-lambda-ssmstringparameter</code> |
|  Java | <code>software.amazon.awsconstructs.services.lambdastringparameter</code> |

Overview

Questo AWS Solutions Construct implementa la funzione AWS Lambda e il parametro AWS Systems Manager Parameter Store String con le autorizzazioni meno privilegiate.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id [string](#)
- props [LambdaToSsmstringparameterProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|-----------------------------|--|--|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| ExistingStringParameterObj? | ssm.StringParameter | Istanza esistente dell'oggetto parametro String SSM, fornendo sia questo che <code>stringParameterProps</code> causerà un errore. |
| StringParameterProps? | ssm.StringParameterProps | L'utente facoltativo ha fornito oggetti di scena per |

| Nome | Tipo | Descrizione |
|---|--------|---|
| | | sovrascrivere gli oggetti di scena predefiniti per il parametro String SSM. Se <code>existingStringParameterObj</code> non è impostato, <code>stringParameterProps</code> è obbligatorio. L'unico supporto a <code>ssm.StringParameterProps.type</code> è <code>STRING</code> se viene fornito un valore diverso, verrà sovrascritto. |
| StringParameterEnvironmentVariableName? | string | Nome facoltativo per la variabile di ambiente del parametro String SSM impostata per la funzione Lambda. |

| Nome | Tipo | Descrizione |
|----------------|------------------------------|--|
| VPC esistente? | ec2.IVpc | <p>Un VPC opzionale esistente in cui deve essere distribuito questo modello. Quando viene distribuita in un VPC, la funzione Lambda utilizzerà ENI nel VPC per accedere alle risorse di rete e verrà creato un endpoint di interfaccia nel parametro VPC per AWS Systems Manager. Se viene fornito un VPC esistente, l'opzione <code>deployVpc</code> non può essere <code>true</code>. Questo utilizza <code>ec2.IVpc</code> per consentire ai clienti di fornire VPC che esistono al di fuori dello stack utilizzando il metodo ec2.Vpc.fromLookup() Metodo.</p> |
| VPCProp? | ec2.VpcProps | <p>Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per il nuovo VPC. <code>enableDnsHostnames</code>, <code>enableDnsSupport</code>, <code>natGatewaySubnetConfiguration</code> sono impostati dal modello, quindi tutti i valori per quelle proprietà fornite qui verranno ignorati. Se <code>deployVpc</code> non è <code>true</code>, questa proprietà verrà ignorata.</p> |

| Nome | Tipo | Descrizione |
|-----------------------------|---------|--|
| DeployVPC? | boolean | <p>Se creare un nuovo VPC basato su <code>vpcProps</code> in cui distribuire questo modello. Impostazione di questo valore su <code>true</code> distribuirà il VPC minimo e più privato per eseguire il pattern:</p> <ul style="list-style-type: none"> • Una sottorete isolata in ogni zona di disponibilità utilizzata dal programma CDK. • <code>enableDnsHostnames</code> e <code>enableDnsSupport</code> saranno entrambi impostati su <code>true</code>. <p>Se questa proprietà è impostata su <code>true</code>, quindi <code>existingVpc</code> non può essere specificato. L'impostazione predefinita è <code>false</code>.</p> |
| StringParameterPermissions? | string | <p>Autorizzazioni facoltative dei parametri String SSM da concedere alla funzione Lambda. Può essere specificato uno degli elementi seguenti: <code>Read</code>, <code>ReadWrite</code> :.</p> |

Proprietà dei modelli

| Nome | Tipo | Descrizione |
|-------------------|-------------------------------------|--|
| LambdaFunction | lambda.Function | Restituisce un'istanza di <code>lambda.Function</code> creato dal costrutto. |
| stringa Parameter | ssm.StringParameter | Restituisce un'istanza di <code>ssm.StringParameter</code> creato dal costrutto. |
| Il VPC? | ec2.IVpc | Restituisce un'interfaccia sul VPC utilizzato dal modello (se presente). Questo può essere un VPC creato dal pattern o dal VPC fornito al costruttore del pattern. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Funzione di AWS Lambda

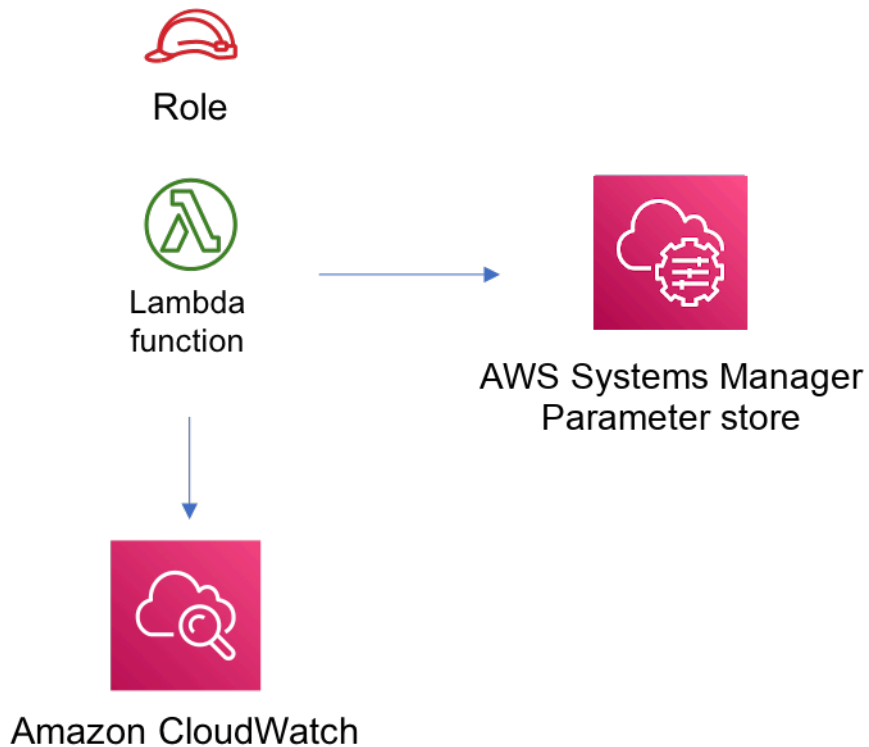
- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Abilitare il tracciamento X-Ray.
- Imposta le variabili di ambiente:
 - `SSM_STRING_PARAMETER_NAME` (predefinito)
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

StringStore parametri di Amazon AWS Systems Manager

- Abilita l'accesso in sola lettura per la funzione AWS Lambda associata.

- Crea un nuovo parametro String SSM con i valori forniti.
- Mantenere il parametro String SSM quando si elimina lo stack CloudFormation.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-constructs/aws-lambda-ssm-stringparameter](https://github.com/aws-solutions-constructs/aws-lambda-ssm-stringparameter)

aws-lambda-sagemakerendpoint

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti all'[controllo](#)

[delle versioni semantiche](#)Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_lambda_sagemakerendpoint</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-lambda-sagemakerendpoint</code> |
|  Java | <code>software.amazon.awsconstructs.services.lambda.sagemakerendpoint</code> |

Overview

Questo AWS Solutions Construct implementa una funzione AWS Lambda collegata a un endpoint Amazon Sagemaker.

Ecco una definizione di modello distribuibile minima in 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);

```

Parametri

- scope [Construct](#)
- id [string](#)
- props [LambdaToSagemakerEndpointProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|---------------------|---------------------------------|---|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |

| Nome | Tipo | Descrizione |
|------------------------------------|--|--|
| LambdafunctionPunps? | <u>lambda.FunctionProps</u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. |
| EsistenteSageMaker endPointOBJ? | <u>sagemaker.CfnEndpoint</u> | Un SageMaker Endpoint opzionale esistente da utilizzare. Fornire sia questo <code>cheEndpointProps</code> causerà un errore. |
| ModelProps? | <u>sagemaker.CfnModel</u> <u>Props</u> any | Proprietà fornite dall'utente per sovrascrivere le proprietà predefinite per il modello SageMaker. Almeno <code>modelProps.primaryContainer</code> deve essere fornito per creare un modello. Per impostazione predefinita, il modello creerà un ruolo con le autorizzazioni minime richieste, ma il client può fornire un ruolo personalizzato con funzionalità aggiuntive e utilizzando <code>modelProps.executionRoleArn</code> . |
| EndPointConfigProps? | <u>sagemaker.CfnEndpoint</u> <u>ConfigProps</u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la configurazione di SageMaker Endpoint. |

| Nome | Tipo | Descrizione |
|----------------|---|---|
| EndPointProps? | <u>sagemaker.CfnEndpointProps</u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per Sagemaker Endpoint. |
| VPC esistente? | <u>ec2.IVpc</u> | Un VPC opzionale esistente in cui deve essere distribuito questo costrutto. Quando viene distribuito in un VPC, la funzione Lambda e Sagemaker Endpoint utilizzeranno le ENI nel VPC per accedere alle risorse di rete. Verrà creato un endpoint di interfaccia nel VPC per Amazon Sagemaker Runtime e Amazon S3 VPC Endpoint. Se viene fornito un VPC esistente, <code>deployVpc</code> non può essere <code>true</code> . |
| VPCProp? | <u>ec2.VpcProps</u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per il nuovo VPC. <code>enableDnsHostnames</code> , <code>enableDnsSupport</code> , <code>natGatewaySubnetConfiguration</code> sono impostati dal costrutto, quindi tutti i valori per quelle proprietà fornite qui verranno ignorati. Se <code>deployVpc</code> non è <code>true</code> , questa proprietà verrà ignorata. |

| Nome | Tipo | Descrizione |
|-----------------------------------|---------|--|
| DeployVPC? | boolean | <p>Se creare un nuovo VPC basato su <code>vpcProps</code> in cui distribuire questo modello. Impostazione di questo valore su <code>true</code> distribuirà il VPC minimo e più privato per eseguire il pattern:</p> <ul style="list-style-type: none"> • Una sottorete isolata in ogni zona di disponibilità utilizzata dal programma CDK. • <code>enableDnsHostnames</code> e <code>enableDnsSupport</code> saranno entrambi impostati su <code>true</code>. <p>Se questa proprietà è impostata su <code>true</code>, quindi <code>existingVpc</code> non può essere specificato. L'impostazione predefinita è <code>false</code>.</p> |
| SageMakerEnvironmentVariableName? | string | Nome facoltativo per la variabile di ambiente endpoint SageMaker impostata per la funzione Lambda. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|--------------------------|--|--|
| LambdaFunction | <u>lambda.Function</u> | Restituisce un'istanza della funzione Lambda creata dal pattern. |
| SageMakerEndpoint | <u>sagemaker.CfnEndpoint</u> | Restituisce un'istanza dell'endpoint di Sagemaker creato dalla serie. |
| SageMakerEndPointConfig? | <u>sagemaker.CfnEndpointConfig</u> | Restituisce un'istanza di SageMaker EndpointConfig creata dal pattern, seexistingSagemakerEndpointObj non viene fornito. |
| SageMakerModel? | <u>sagemaker.CfnModel</u> | Restituisce un'istanza del modello Sagemaker creato dalla serie, seexistingSagemakerEndpointObj non viene fornito. |
| Il VPC? | ec2.IVpc | Restituisce un'istanza del VPC creato dal pattern, sedeployVpc ètrue, o seexistingVpc viene fornito. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

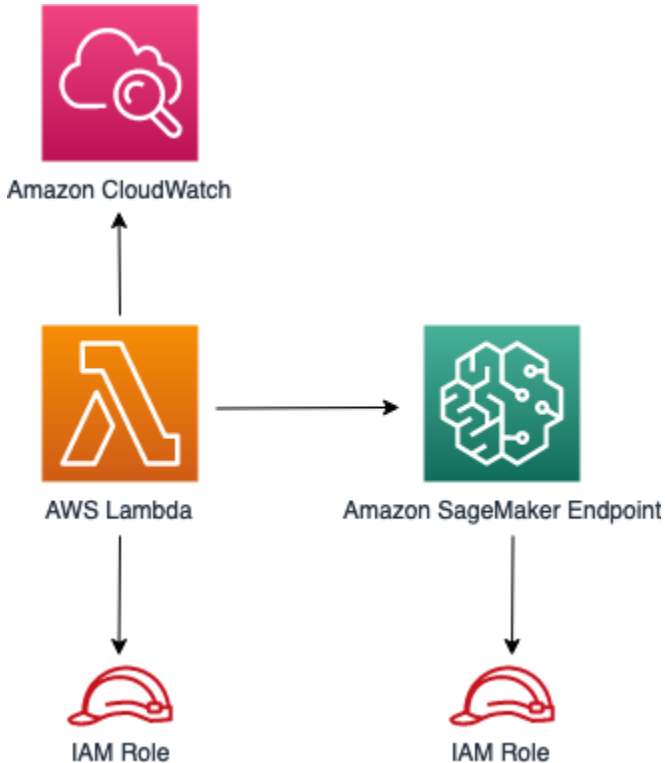
Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Consentire alla funzione di richiamare l'endpoint di Sagemaker per Inferenze.
- Configurare la funzione per accedere alle risorse nel VPC, in cui viene distribuito l'endpoint Sagemaker.
- Attivare il tracciamento X-Ray.
- Impostazione delle variabili di ambiente:
 - SAGEMAKER_ENDPOINT_NAME (predefinito)
 - AWS_NODEJS_CONNECTION_REUSE_ENABLED(per le funzioni Node 10.x e successive)

Endpoint SageMaker

- Configurare privilegi limitati per creare risorse Sagemaker.
- Distribuire il modello Sagemaker, EndPointConfig ed endpoint.
- Configurare l'endpoint Sagemaker da distribuire in un VPC.
- Distribuire S3 VPC Endpoint e Sagemaker Runtime Interfaccia VPC.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:






[@aws -solutions-constructs/aws-lambda-sagemakerendpoint](https://github.com/aws-solutions-constructs/aws-lambda-sagemakerendpoint)

aws-lambda-secretsmanager

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti all'[controllo delle versioni semantico](#)Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_lambda_secretsmanager</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-lambda-secretsmanager</code> |
|  Java | <code>software.amazon.awsconstructs.services.lambda_secretsmanager</code> |

Overview

Questo AWS Solutions Construct implementa la funzione AWS Lambda e il segreto AWS Secrets Manager con le autorizzazioni meno privilegiate.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id `string`
- props [LambdaToSecretsmanagerProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--|---|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | L'utente ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per la funzione Lambda. |
| Secretpuntelli? | secretsmanager.SecretProps | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per Secrets Manager. |
| EsistingSecretoBJ? | secretsmanager.Secret | Istanza esistente di Secrets Manager oggetto segreto, Se questo è impostato, la |

| Nome | Tipo | Descrizione |
|---|---------------------------------------|---|
| | | proprietà <code>secretProps</code> viene ignorato. |
| <code>GrantWriteAccess?</code> | <code>boolean</code> | Accesso facoltativo in scrittura al segreto per la funzione Lambda (di sola lettura per impostazione predefinita). |
| <code>SecretEnvironmentVariableName?</code> | <code>string</code> | Nome facoltativo per la variabile di ambiente segreto Secrets Manager impostata per la funzione Lambda. |
| VPC esistente? | <code>ec2.IVpc</code> | Un VPC opzionale esistente in cui deve essere distribuito questo modello. Quando viene distribuita in un VPC, la funzione Lambda utilizzerà ENI nel VPC per accedere alle risorse di rete e verrà creato un endpoint di interfaccia nel VPC per AWS Secrets Manager. Se viene fornito un VPC esistente, <code>deployVpc</code> non può essere <code>true</code> : . Questo utilizza <code>ec2.IVpc</code> per consentire ai client di fornire i VPC che esistono al di fuori dello stack utilizzando il <code>ec2.Vpc.fromLookup()</code> Metodo. |

| Nome | Tipo | Descrizione |
|----------|------------------------------|---|
| VPCProp? | ec2.VpcProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per il nuovo VPC. <code>enableDns Hostnames</code> , <code>enableDns Support</code> , <code>natGateways</code> , <code>subnetConfigurations</code> sono impostati dal modello, quindi tutti i valori per quelle proprietà fornite qui verranno ignorati. <code>SeedDeployVpc</code> non è <code>true</code> questa proprietà verrà ignorata. |

| Nome | Tipo | Descrizione |
|------------|---------|---|
| DeployVPC? | boolean | <p>Se creare un nuovo VPC basato su <code>vpcProps</code> in cui distribuire questo modello. Impostazione di questo valore su <code>true</code> distribuirà il VPC minimo e più privato per eseguire il pattern:</p> <ul style="list-style-type: none"> • Una sottorete isolata in ciascuna zona di disponibilità utilizzata dal programma CDK • <code>enableDnsHostnames</code> e <code>enableDnsSupport</code> saranno entrambi impostati su <code>true</code> <p>Se questa proprietà è <code>true</code>, quindi <code>existingVpc</code> non può essere specificato. L'impostazione predefinita è <code>false</code>.</p> |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------|--|--|
| LambdaFunction | <u>lambda.Function</u> | Restituisce un'istanza di <code>lambda.Function</code> creato dal costrutto. |
| Secret | <u>secretsmanager.Secret</u> | Restituisce un'istanza di <code>secretsmanager.Secret</code> creato dal costrutto. |

| Nome | Tipo | Descrizione |
|------|--------------------------|--|
| VPC? | ec2.IVpc | Restituisce un'interfaccia sul VPC utilizzato dal modello (se presente). Questo può essere un VPC creato dal pattern o dal VPC fornito al costruttore del pattern. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni impofterà i seguenti valori predefiniti:

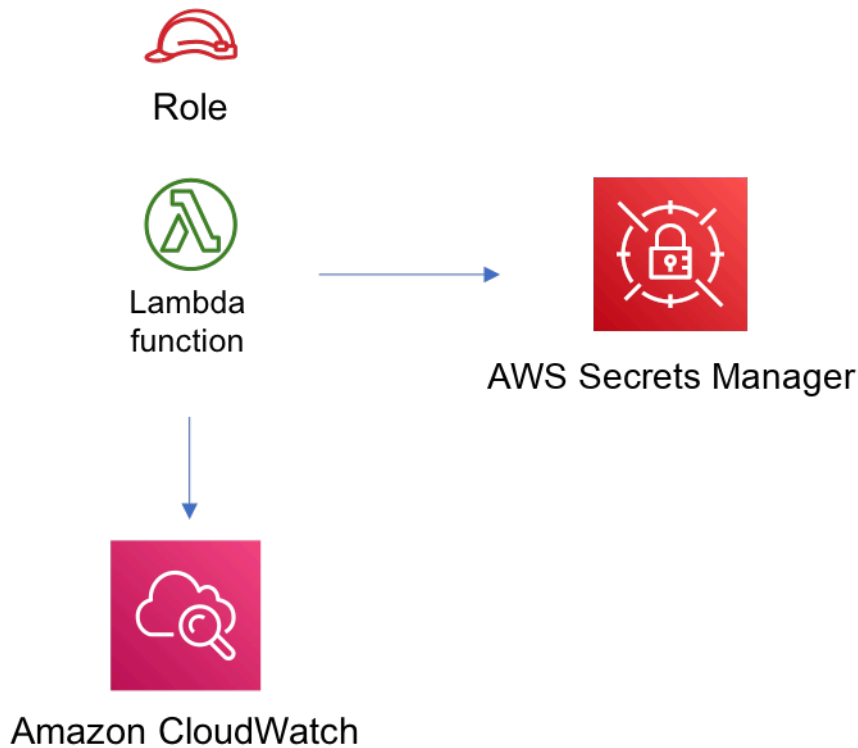
Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Abilitare il tracciamento X-Ray.
- Impostazione delle variabili di ambiente:
 - (default) SECRET_ARN contenente l'ARN del segreto come restituito da CDK [SecretArn](#) property
 - AWS_NODEJS_CONNECTION_REUSE_ENABLED(per le funzioni Node 10.x e successive)

Segreto di Amazon Secrets Manager

- Abilita l'accesso in sola lettura per la funzione AWS Lambda associata
- Abilitare la crittografia lato server utilizzando una chiave del servizio di gestione delle chiavi predefinita per l'account e l'area
- Crea un nuovo segreto:
 - Nome casuale (predefinito)
 - (default) valore casuale
- Mantieni il segreto durante l'eliminazione dello stack CloudFormation

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws-solutions-constructs/aws-lambda-secretsmanager](https://github.com/@aws-solutions-constructs/aws-lambda-secretsmanager)

aws-lambda-sns

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic version \(Controllo versioni\)](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_lambda_sns</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-lambda-sns</code> |
|  Java | <code>software.amazon.awsconstructs.services.lambdasns</code> |

Overview

Questo AWS Solutions Construct implementa una funzione AWS Lambda collegata a un argomento Amazon SNS.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id [string](#)
- props [LambdaToSnsProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--------------------------------------|--|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| Esistente Topicobj? | sns.Topic | Istanza esistente dell'oggetto argomento SNS, fornendo sia questo che <code>topicProps</code> causerà un errore. |
| TopicProps? | sns.TopicProps | L'utente facoltativo ha fornito proprietà per ignorare le |

| Nome | Tipo | Descrizione |
|----------------|--------------------------|--|
| | | proprietà predefinite per l'argomento SNS. |
| VPC esistente? | ec2.IVpc | <p>Un VPC opzionale esistente in cui deve essere distribuito questo modello. Quando viene distribuita in un VPC, la funzione Lambda utilizzerà ENI nel VPC per accedere alle risorse di rete e verrà creato un endpoint di interfaccia nel VPC per Amazon SQS. Se viene fornito un VPC esistente, <code>deployVpc</code> non può essere <code>true</code>. Questo utilizza <code>ec2.IVpc</code> per consentire ai clienti di fornire i VPC che esistono al di fuori dello stack utilizzando il ec2.Vpc.fromLookup() Metodo.</p> |

| Nome | Tipo | Descrizione |
|------------|---------|--|
| DeployVPC? | boolean | <p>Se creare un nuovo VPC basato su <code>vpcProps</code> in cui distribuire questo modello. Impostazione di questo valore su <code>true</code> distribuirà il VPC minimo e più privato per eseguire il pattern:</p> <ul style="list-style-type: none">• Una sottorete isolata in ogni zona di disponibilità utilizzata dal programma CDK.• <code>enableDnsHostnames</code> e <code>enableDnsSupport</code> saranno entrambi impostati su <code>true</code>. <p>Se questa proprietà è <code>true</code>, quindi <code>existingVpc</code> non può essere specificato. L'impostazione predefinita è <code>false</code>.</p> |

| Nome | Tipo | Descrizione |
|-----------------------------------|-------------------------------------|--|
| VPCProp? | <u>ec2.VpcProps</u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per il nuovo VPC. <code>enableDns Hostnames</code> , <code>enableDns Support</code> , <code>natGateways</code> , <code>esubnetConfiguration</code> sono impostati dal modello, quindi tutti i valori per quelle proprietà fornite qui verranno ignorati. Se <code>deployVpc</code> non è <code>true</code> , questa proprietà verrà ignorata. |
| TopArnEnvironmentVariableName? | string | Nome facoltativo per la variabile di ambiente ARN argomento SNS impostata per la funzione Lambda. |
| TopicNameEnvironmentVariableName? | string | Nome facoltativo per la variabile di ambiente nome argomento SNS impostata per la funzione Lambda. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------|--|--|
| LambdaFunction | <u>lambda.Function</u> | Restituisce un'istanza della funzione Lambda creata dal pattern. |

| Nome | Tipo | Descrizione |
|----------|---------------------------|--|
| SNSTopic | sns.Topic | Restituisce un'istanza dell'argomento SNS creato dal modello. |
| VPC? | ec2.IVpc | Restituisce un'istanza del VPC utilizzato dal modello (se presente). Questo può essere un VPC creato dal pattern o dal VPC fornito al costruttore del pattern. |

Impostazioni predefinite

L'implementazione predefinita del Costrutto senza alcun override imposterà i seguenti valori predefiniti:

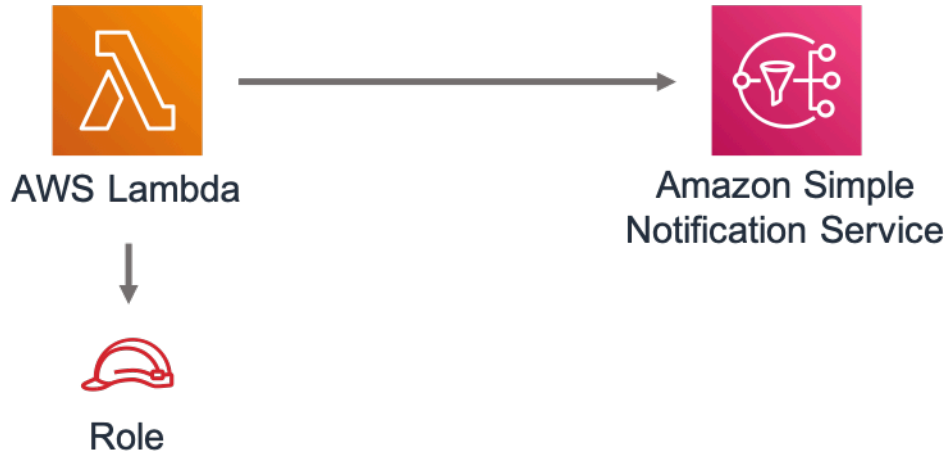
Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Abilitare il tracciamento X-Ray
- Imposta le variabili di ambiente:
 - SNS_TOPIC_NAME (predefinito)
 - SNS_TOPIC_ARN (predefinito)
 - AWS_NODEJS_CONNECTION_REUSE_ENABLED(per le funzioni Node 10.x e successive)

Argomenti Amazon SNS

- Configurare le autorizzazioni di accesso con privilegi minimi per l'argomento SNS.
- Abilitare la crittografia lato server utilizzando la chiave AWS gestita.
- Applicazione della crittografia dei dati in transito

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:





[@aws-solutions-constructs/aws-lambda-sns](https://github.com/aws-solutions-constructs/aws-lambda-sns)

aws-lambda-sqs

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti all'[controllo delle versioni semantiche](#)Modello: Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_lambda_sqs</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-lambda-sqs</code> |
|  Java | <code>software.amazon.awsconstructs.services.lambdasqs</code> |

Overview

Questo AWS Solutions Construct implementa una funzione AWS Lambda collegata a una coda Amazon SQS.

Ecco una definizione di modello distribuibile minima in 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);
```


Parametri

- `scope` [Construct](#)
- `idstring`
- `props` [LambdaToSqsProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|-----------------------------------|--------------------------------------|--|
| <code>EsistenteLambdaobj?</code> | lambda.Function | Una funzione Lambda opzionale esistente da utilizzare al posto della funzione predefinita. Fornire sia questo che <code>lambdaFunctionProps</code> . In caso contrario, si verifica un errore. |
| <code>LambdafunctionPunps?</code> | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. |
| <code>ExistingQueueObj?</code> | sqs.Queue | Una coda SQS esistente facoltativa da utilizzare al posto della coda predefinita. Fornire sia questo che <code>queueProps</code> . In caso contrario, si verifica un errore. |
| <code>QueueProps?</code> | sqs.QueueProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la coda SQS. |

| Nome | Tipo | Descrizione |
|------------------------|--------------------------------|--|
| EnableQueuePurging? | boolean | Indica se concedere autorizzazioni aggiuntive alla funzione Lambda che consente di eliminare la coda SQS. L'impostazione predefinita è false. |
| DeployDeadLetterQueue? | boolean | Se creare una coda secondaria da utilizzare come coda DLQ. L'impostazione predefinita è true. |
| DeadLetterQueueProps? | sqs.QueueProps | Elementi di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per la coda delle lettere morte. Utilizzato solo se il valore <code>deployDeadLetterQueue</code> è impostato su true. |
| MaxReceiveCount? | number | Il numero di volte in cui un messaggio può essere rimosso senza successo prima di essere spostato nella coda DLQ. L'impostazione predefinita è 15. |

| Nome | Tipo | Descrizione |
|----------------|--------------------------|--|
| VPC esistente? | ec2.IVpc | <p>Un VPC opzionale esistente in cui deve essere distribuito questo modello. Quando viene distribuita in un VPC, la funzione Lambda utilizzerà ENI nel VPC per accedere alle risorse di rete e verrà creato un endpoint di interfaccia nel VPC per Amazon SQS. Se viene fornito un VPC esistente, <code>deployVpc</code> non può essere <code>true</code>: <code>ec2.IVpc</code> viene utilizzato per consentire ai clienti di fornire i VPC che esistono al di fuori dello stack utilizzando il metodo <code>ec2.Vpc.fromLookup()</code>.</p> |

| Nome | Tipo | Descrizione |
|------------|---------|--|
| DeployVPC? | boolean | <p>Se creare un nuovo VPC basato su <code>vpcProps</code> in cui distribuire questo modello. Impostazione di questo limite su <code>true</code> distribuirà il VPC minimo e più privato per eseguire il pattern:</p> <ul style="list-style-type: none">• Una sottorete isolata in ciascuna zona di disponibilità utilizzata dal programma CDK• <code>enableDnsHostnames</code> e <code>enableDnsSupport</code> saranno entrambi impostati su <code>true</code> <p>Se questa proprietà è <code>true</code>, quindi <code>existingVpc</code> non può essere specificato. L'impostazione predefinita è <code>false</code>.</p> |

| Nome | Tipo | Descrizione |
|-------------------------------|------------------------------|---|
| VPCProp? | ec2.VpcProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per il nuovo VPC. <code>enableDnsHostnames</code> , <code>enableDnsSupport</code> , <code>natGateways</code> , <code>subnetConfigurations</code> sono impostati dal modello, quindi tutti i valori per quelle proprietà fornite qui verranno sovrascritti. <code>SeedployVpc</code> non è <code>true</code> , questa proprietà verrà ignorata. |
| QueueEnvironmentVariableName? | string | Nome facoltativo per la variabile di ambiente URL della coda SQS impostata per la funzione Lambda. |

Proprietà del modello

| Nome | Tipo | Descrizione |
|------------------|---------------------------------|---|
| DeadLetterQueue? | sqs.Queue | Restituisce un'istanza della coda di lettere morte creata dal modello, se ne viene distribuita una. |
| LambdaFunction | lambda.Function | Restituisce un'istanza della funzione Lambda creata dal pattern. |
| SQSqueue | sqs.Queue | Restituisce un'istanza della coda SQS creata dal pattern. |

| Nome | Tipo | Descrizione |
|------|--------------------------|--|
| VPC? | ec2.IVpc | Restituisce un'istanza del VPC creato o utilizzato dal pattern (se presente). Questo può essere un VPC creato dal modello o un VPC fornito al costruttore del pattern. |

Impostazioni predefinite

L'implementazione predefinita del Costrutto senza alcun override imposterà i seguenti valori predefiniti:

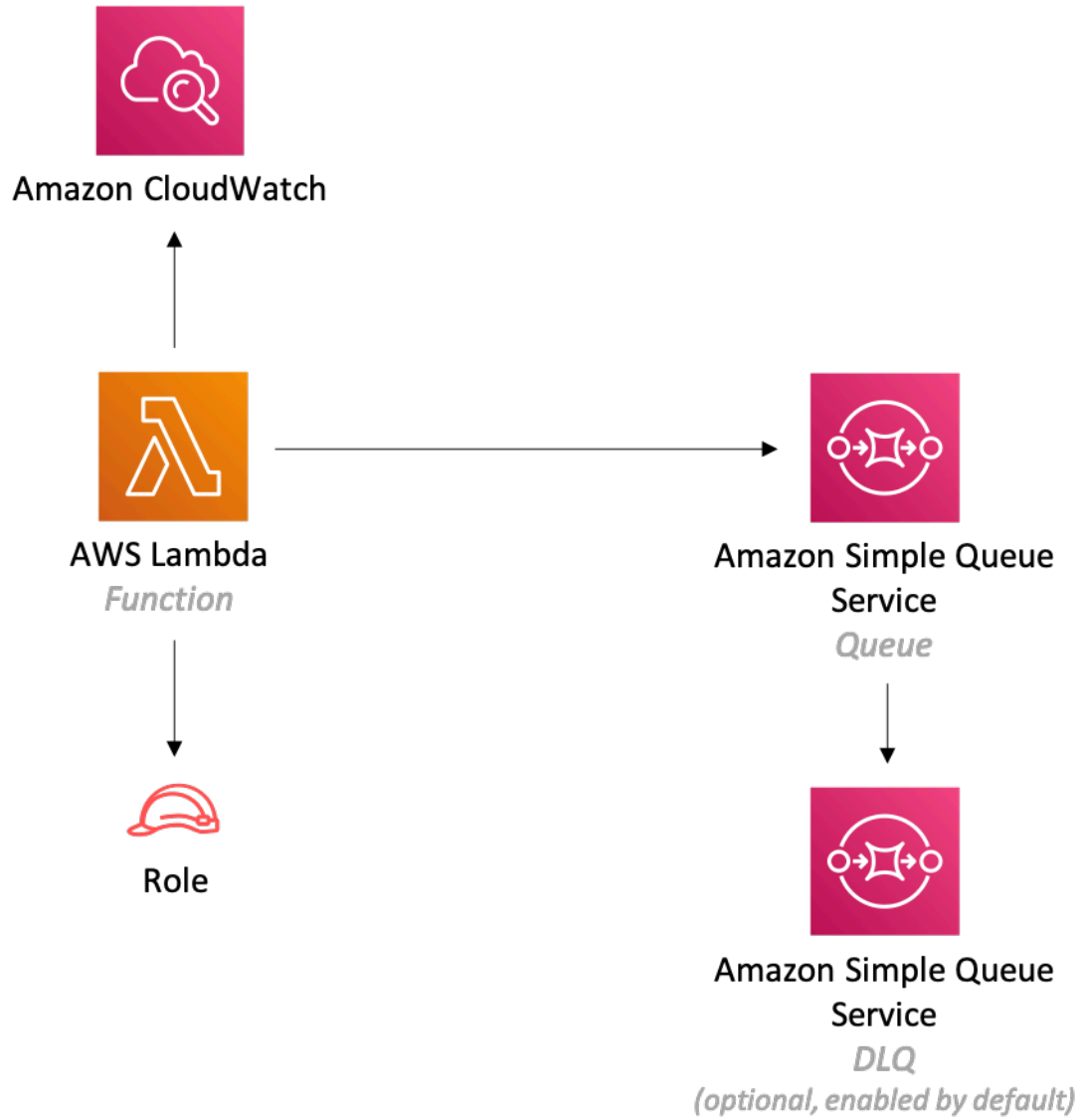
Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Consenti alla funzione di inviare solo messaggi alla coda (l'eliminazione può essere abilitata usando il comando `enableQueuePurge`).
- Abilitare il rilevamento di X-Ray
- Impostazione delle variabili di ambiente:
 - `SQS_QUEUE_URL`
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Coda Amazon SQS

- Distribuire la coda DLQ SQS per la coda SQS di origine.
- Abilitare la crittografia lato server per la coda SQS di origine tramite la chiave AWS gestita.
- Applicazione della crittografia dei dati in transito

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-buils/aws-lambda-sqs](#)




aws-lambda-sqs-lambda

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti all'[controllo delle versioni semantico](#)Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_lambda_sqs_lambda</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-lambda-sqs-lambda</code> |
|  Java | <code>software.amazon.awsconstructs.services.lambdasqslambda</code> |

Overview

Questo modello AWS Solutions Constructs implementa (1) una funzione AWS Lambda configurata per inviare messaggi a una coda; (2) una coda Amazon SQS; e (3) una funzione AWS Lambda configurata per consumare messaggi dalla coda.

Ecco una definizione di modello distribuibile minima in 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);

```

Parametri

- scope [Construct](#)
- id [string](#)
- props [LambdaToSqsToLambdaProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|-----------------------------|---------------------------------|---|
| EsistenteProducerLambdaobj? | lambda.Function | Funzione Lambda opzionale esistente da utilizzare al posto della funzione predefinita per l'invio di messaggi alla coda. Fornire sia questo che <code>producerLambdaFunc</code> |

| Nome | Tipo | Descrizione |
|---|--|--|
| | | <code>tionProps</code> In caso contrario, si verifica un errore |
| <code>ProducerLambdafunctionProps?</code> | <u><code>lambda.FunctionProps</code></u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda del produttore. |
| <code>ExistingQueueObj?</code> | <u><code>sqs.Queue</code></u> | Una coda SQS esistente facoltativa da utilizzare al posto della coda predefinita. Fornire sia questo <code>chequeueProps</code> In caso contrario, si verifica un errore |
| <code>QueueProps?</code> | <u><code>sqs.QueueProps</code></u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la coda SQS. Fornire sia questo <code>cheexistingQueueObj</code> In caso contrario, si verifica un errore |
| <code>DeployDeadLetterQueue?</code> | <code>boolean</code> | Indica se creare una coda secondaria da utilizzare come coda DEFAULT. L'impostazione predefinita è <code>true</code> . |

| Nome | Tipo | Descrizione |
|------------------------------|---|---|
| DeadLetterQueueProps? | <u>sqs.QueueProps</u> | Elementi di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per la coda delle lettere morte. Utilizzato solo se il valore <code>deployDeadLetterQueue</code> La proprietà viene impostata su <code>true</code> . |
| MaxReceiveCount? | <code>number</code> | Il numero di volte in cui un messaggio può essere rimosso senza successo prima di essere spostato nella coda LQ. L'impostazione predefinita è 15. |
| EsistenteConsumerLambdaobj? | <u>lambda.Function</u> | Funzione Lambda opzionale esistente da utilizzare al posto della funzione predefinita per ricevere o consumare messaggi dalla coda. Fornire sia questo che <code>consumerLambdaFunctionProps</code> . In caso contrario, si verifica un errore. |
| ConsumerLambdafunctionProps? | <u>lambda.FunctionProps</u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda consumer. |

| Nome | Tipo | Descrizione |
|-------------------------------|--------|---|
| QueueEnvironmentVariableName? | string | Nome facoltativo per la variabile di ambiente URL della coda SQS impostata per la funzione Lambda del produttore. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|------------------------|---------------------------------|---|
| ConsumerLambdafunction | lambda.Function | Restituisce un'istanza della funzione Lambda consumer creata dal modello. |
| DeadLetterQueue? | sqs.Queue | Restituisce un'istanza della coda di lettere morte creata dal modello, se ne viene distribuita una. |
| ProducerLAMBDAFUNZIONE | lambda.Function | Restituisce un'istanza della funzione Lambda del produttore e creata dal modello. |
| SQSqueue | sqs.Queue | Restituisce un'istanza della coda SQS creata dal pattern. |

Impostazioni predefinite

L'implementazione predefinita di questo Costrutto (senza proprietà sovrascritte) adatterà ai seguenti valori predefiniti:

Funzioni AWS Lambda

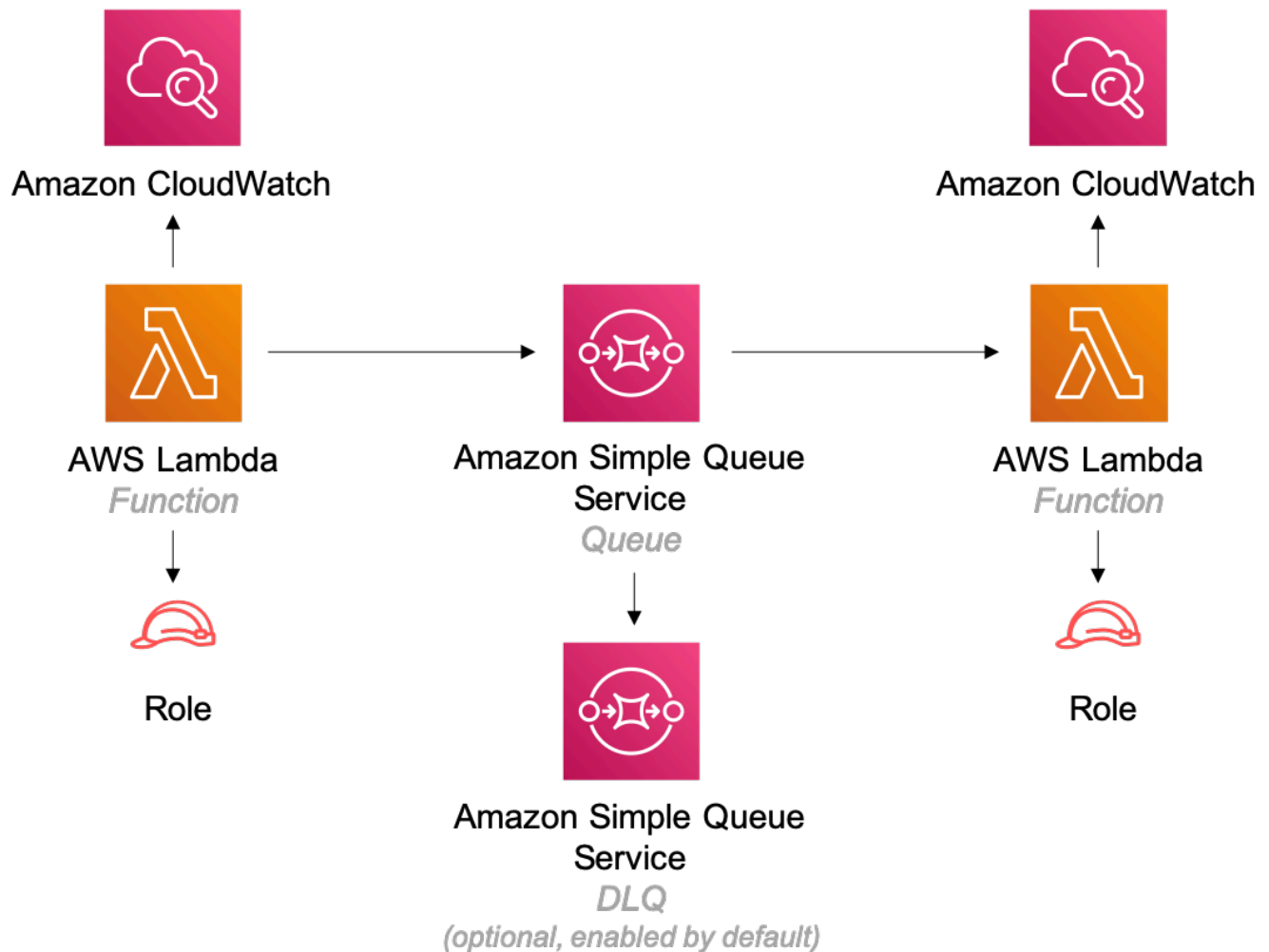
- Configurare il ruolo IAM di accesso con privilegi limitati per le funzioni Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per le funzioni NodeJS Lambda.

- Abilitare il monitoraggio di X-Ray
- Impostazione delle variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Coda Amazon SQS

- Distribuire una coda di origine secondaria per la coda primaria.
- Abilitare la crittografia lato server per la coda primaria utilizzando una chiave AWS gestita.
- Applicazione della crittografia dei dati in transito

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -soluzioni-costruzioni/aws-lambda-sqs-lambda](https://github.com/aws-soluzioni-costruzioni/aws-lambda-sqs-lambda)


aws-lambda-step-funzione

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic Versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_lambda_step_function</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-lambda-step-function</code> |
|  Java | <code>software.amazon.awsconstructs.services.lambda-step-function</code> |

Overview

Questo AWS Solutions Construct implementa una funzione AWS Lambda collegata a una funzione Step AWS.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id [string](#)
- props [LambdaToStepFunctionProps](#)

Schema di costruzione

| Nome | Tipo | Descrizione |
|-------------------------------------|---|--|
| EsistenteLambdaobj? | <u>lambda.Function</u> | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | <u>lambda.FunctionProps</u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| StateMachineProps | <u>sfn.StateMachinePr ops</u> | L'utente ha fornito oggetti di scena per la SFN.State Machine. |
| CreateCloudWatchalarms | boolean | Indica se creare allarmi CloudWatch consigliati. |
| LogGroupProps? | <u>logs.LogGroupProps</u> | Opzionali forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatch Logs. |
| StateMachineEnvironmentVariableName | string | Nome facoltativo per la variabile di ambiente della macchina dello stato Step Functions impostata per la funzione Lambda del produttore. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------------|---|--|
| CloudWatchArms? | <u>cloudwatch.Alarm[]</u> | Restituisce un elenco di uno o più avvisi CloudWatch creati dal modello. |
| LambdaFunction | <u>lambda.Function</u> | Restituisce un'istanza della funzione Lambda creata dal pattern. |
| StateMachine | <u>sfn.StateMachine</u> | Restituisce un'istanza della macchina a stati creata dal pattern. |
| StateMachineLogGroup | <u>logs.LogGroup</u> | Restituisce un'istanza del gruppo di log creato dal pattern per la macchina a stati. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

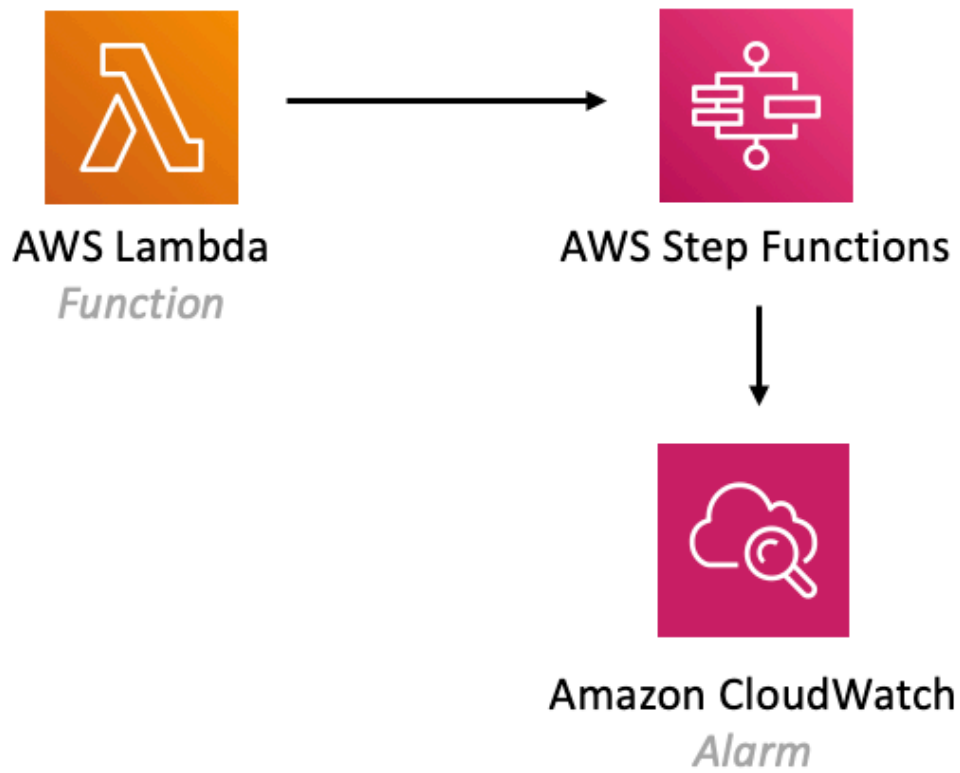
Funzione di AWS Lambda

- Configurare un ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per le funzioni NodeJS Lambda.
- Attivare il tracciamento X-Ray.
- Impostare le variabili di ambiente:
 - STATE_MACHINE_ARN (predefinito)
 - AWS_NODEJS_CONNECTION_REUSE_ENABLED(per le funzioni Node 10.x e successive)

AWS Step Functions Machine

- Distribuisci gli allarmi CloudWatch best practice per il AWS Step Functions State Machine.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-constructs/aws-lambda-step-function](https://github.com/aws-solutions-constructs/aws-lambda-step-function)


aws-s3-lambda

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti all'[controllo delle versioni semantiche](#)Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_s3_lambda</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-s3-lambda</code> |
|  Java | <code>software.amazon.awsconstructs.services.s3lambda</code> |

Overview

Questo AWS Solutions Construct implementa un bucket Amazon S3 collegato a una funzione AWS Lambda.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id `string`
- props [S3ToLambdaProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--------------------------------------|--|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| Esistente BucketObj? | s3.Bucket | Istanza esistente dell'oggetto S3 Bucket. Se questo è fornito, quindi anche <code>provideBucketProps</code> è un errore. |

| Nome | Tipo | Descrizione |
|---------------------|---|---|
| Bucket Props? | <u>s3.BucketProps</u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per il bucket. Ignorato se <code>unexistingBucketObj</code> viene fornito. |
| S3EventSourceProps? | <u>S3EventSourceProps</u> | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per <code>S3EventSourceProps</code> |

Proprietà dei modelli

| Nome | Tipo | Descrizione |
|------------------|--|---|
| LambdaFunction | <u>lambda.Function</u> | Restituisce un'istanza della funzione Lambda creata dal pattern. |
| S3Bucket? | <u>s3.Bucket</u> | Restituisce un'istanza del bucket S3 creato dalla serie. |
| S3loggingBucket? | <u>s3.Bucket</u> | Restituisce un'istanza del bucket di registrazione creato dal pattern per il bucket S3. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

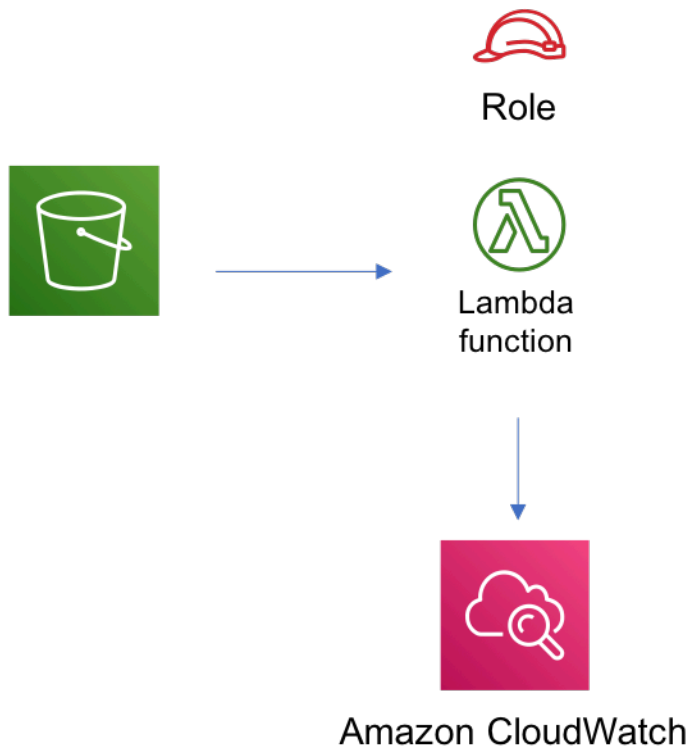
Amazon S3

- Configurare la registrazione di accesso per il bucket S3
- Abilita la crittografia lato server per S3 Bucket utilizzando la chiave AWS gestita KMS.
- Attivare il controllo delle versioni per S3 Bucket.
- Non consentire l'accesso pubblico per S3 Bucket.
- Mantenere il bucket S3 quando si elimina lo stack CloudFormation.
- Applicazione della crittografia dei dati in transito
- Applica la regola del ciclo di vita per spostare le versioni degli oggetti non correnti nell'archivio Glacier dopo 90 giorni.

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Attivare il tracciamento X-Ray
- Impostazione delle variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-buils/aws-s3-lambda](#)

aws-s3-sqs

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti all'[controllo delle versioni semantico](#)Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_s3_sqs</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-s3-sqs</code> |
|  Java | <code>software.amazon.awsconstructs.services.s3sqs</code> |

Overview

Questo Construct di soluzioni AWS implementa un bucket Amazon S3 configurato per inviare notifiche a una coda Amazon SQS.

Ecco una definizione di modello distribuibile minima in TypeScript:

```
import { S3ToSqs } from "@aws-solutions-constructs/aws-s3-sqs";  
  
new S3ToSqs(stack, 'S3ToSQSPattern', {});
```

Initializer

```
new S3ToSqs(scope: Construct, id: string, props: S3ToSqsProps);
```

Parametri

- `scope` [Construct](#)

- `idstring`
- `props` [S3ToSqsProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--|---|
| Esistente BucketObj? | s3.Bucket | Istanza esistente dell'oggetto S3 Bucket. Se questo è fornito, quindi anche <code>provideBucketProps</code> è un errore. |
| Bucket Props? | s3.BucketProps | Oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il bucket S3. |
| S3EventTypes? | s3.EventType[] | I tipi di evento S3 che attiveranno la notifica. L'impostazione predefinita è <code>s3.EventType.OBJECT_CREATED</code> . |
| S3EventFilters? | s3.NotificationKeyFilter[] | Le regole filtro chiave oggetto S3 per determinare quali oggetti attivano questo evento. Se non viene specificato, non verranno applicate regole di filtro. |
| ExistingQueueObj? | sqs.Queue | Una coda SQS esistente facoltativa da utilizzare al posto della coda predefinita. Fornire sia questo <code>queueProps</code> causerà un errore. Se la coda SQS è crittografata, la chiave del |

| Nome | Tipo | Descrizione |
|------------------------|--------------------------------|---|
| | | Servizio di gestione delle chiavi utilizzata per la crittografia deve essere un CMK gestito dal cliente. |
| QueueProps? | sqs.QueueProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la coda SQS. Ignorato se <code>unexistingQueueObj</code> viene fornito. |
| DeadLetterQueueProps? | sqs.QueueProps | Oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per la coda delle lettere morte. Utilizzato solo se il valore <code>deployDeadLetterQueue</code> viene impostata su <code>true</code> . |
| DeployDeadLetterQueue? | <code>boolean</code> | Indica se creare una coda secondaria da utilizzare come coda SDQ. L'impostazione predefinita è <code>true</code> . |
| MaxReceiveCount? | <code>number</code> | Il numero di volte in cui un messaggio può essere rimosso senza successo prima di essere spostato nella coda SDQ. L'impostazione predefinita è 15. |

| Nome | Tipo | Descrizione |
|---|------------------------------|---|
| EnableEncryptionWithCustomerManagedKey? | boolean | Indica se utilizzare una chiave del servizio di gestione delle chiavi, gestita da questa app CDK o importata. Se si importa una chiave di crittografia, questa deve essere specificata nella casella <code>encryptionKey</code> per questo costrutto. |
| encryptionKey? | kms.Key | Chiave di crittografia opzionale esistente da utilizzare al posto della chiave di crittografia predefinita. |
| EncryptionKeyProps? | kms.KeyProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la chiave di crittografia. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|------------------|---------------------------|---|
| SQSqueue | sqs.Queue | Restituisce un'istanza della coda SQS creata dal pattern. |
| DeadLetterQueue? | sqs.Queue | Restituisce un'istanza della coda di lettere morte creata dal modello, se ne viene distribuita una. |

| Nome | Tipo | Descrizione |
|------------------|---------------------------|---|
| encryptionKey | kms.IKey | Restituisce un'istanza della chiave di crittografia creata dal modello. |
| S3Bucket? | s3.Bucket | Restituisce un'istanza del bucket S3 creato dalla serie. |
| S3loggingBucket? | s3.Bucket | Restituisce un'istanza del bucket di registrazione creato dal pattern per il bucket S3. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imporrà i seguenti valori predefiniti:

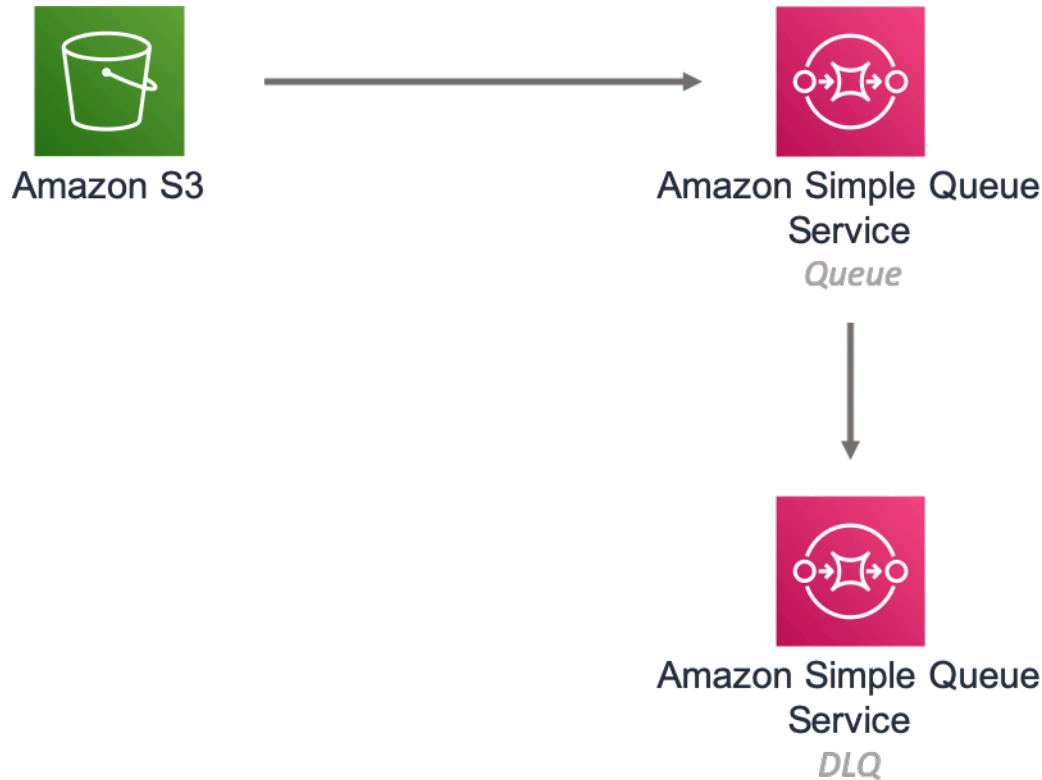
Amazon S3

- Configurare la registrazione degli accessi per S3
- Abilitare la crittografia lato server per S3 Bucket utilizzando la chiave AWS gestita
- Attivare il controllo delle versioni per S3 Bucket
- Non consentire l'accesso pubblico per S3 Bucket
- Mantenere il bucket S3 quando si elimina lo stack CloudFormation
- Applicazione della crittografia dei dati in transito
- Applica la regola del ciclo di vita per spostare versioni degli oggetti non correnti nell'archivio Glacier dopo 90 giorni

Coda Amazon SQS

- Configurare autorizzazioni di accesso con privilegi minimi per SQS Queue
- Distribuire la coda SQS
- Abilitare la crittografia lato server per SQS utilizzando la chiave del servizio di gestione delle chiavi gestita dal cliente
- Applicazione della crittografia dei dati in transito

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:






[@aws-solutions-buils/aws-s3-sqs](https://github.com/aws-solutions-buils/aws-s3-sqs)

aws-s3-step-funzione

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Controllo delle versioni semantiche](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_s3_step_function</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-s3-step-function</code> |
|  Java | <code>software.amazon.awsconstructs.services.s3stepfunction</code> |

Overview

Questo AWS Solutions Construct implementa un bucket Amazon S3 collegato a una funzione Step AWS.

Note

Questo costrutto utilizza Amazon EventBridge (Amazon CloudWatch Events) per attivare AWS Step Functions. EventBridge è più flessibile, ma l'attivazione di Step Functions con S3 Event Notifications ha meno latenza ed è più conveniente. Se il costo e/o la latenza sono un problema, è necessario considerare la distribuzione `aws-s3-lambda-aws-lambda-stepfunctions` al posto di questo costrutto.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id [string](#)
- props [S3ToStepFunctionProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--------------------------------|---|
| Esistente BucketObj? | s3.IBucket | Istanza esistente dell'oggetto S3 Bucket. Se questo è fornito, quindi anche <code>provideBucketProps</code> è un errore. |
| Bucket Props? | s3.BucketProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per il bucket. Ignorato se <code>unexistingBucketObj</code> viene fornito. |

| Nome | Tipo | Descrizione |
|------------------------|--|--|
| StateMachineProps | sfn.StateMachinePr ops | L'utente facoltativo ha fornito oggetti di scena per sovrascrivere gli oggetti di scena predefiniti per SFN.State Machine. |
| EventRuleProps? | events.RuleProps | L'utente facoltativo ha fornito EventRuleProps per sovrascrivere i valori predefiniti. |
| DeployCloudTrail? | boolean | Se distribuire un Trail in AWS CloudTrail per registrare gli eventi API in Amazon S3. L'impostazione predefinita è true. |
| CreateCloudWatchalarms | boolean | Indica se creare allarmi CloudWatch consigliati. |
| LogGroupProps? | logs.LogGroupProps | Elementi di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per il gruppo di log CloudWatch Logs. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|-------------------|----------------------------------|--|
| Cloudtrail? | cloudtrail.Trail | Restituisce un'istanza del percorso Cloudtrail creato dal pattern. |
| CloudTrailBucket? | s3.Bucket | Restituisce un'istanza del bucket creato dal pattern per |

| Nome | Tipo | Descrizione |
|--------------------------|------------------------------------|--|
| | | la memorizzazione dei dati di traccia Cloudtrail. |
| CloudTrailLoggingBucket? | s3.Bucket | Restituisce un'istanza del bucket di registrazione creato dal pattern per il bucket principale utilizzato dal percorso Cloudtrail. |
| CloudWatchArms? | cloudwatch.Alarm[] | Restituisce un elenco di uno o più allarmi CloudWatch creati dal pattern. |
| S3Bucket? | s3.Bucket | Restituisce un'istanza del bucket S3 creato dalla serie. |
| S3loggingBucket? | s3.Bucket | Restituisce un'istanza del bucket di registrazione creato dal pattern per il bucket S3. |
| StateMachine | sfn.StateMachine | Restituisce un'istanza della macchina a stati creata dal pattern. |
| StateMachineLogGroup | logs.LogGroup | Restituisce un'istanza del gruppo di log creato dal pattern per la macchina a stati. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

Amazon S3

- Configurare la registrazione accesso per S3 Bucket
- Abilita la crittografia lato server per S3 Bucket utilizzando la chiave AWS gestita KMS.

- Attivare il controllo delle versioni per S3 Bucket.
- Non consentire l'accesso pubblico per S3 Bucket.
- Mantenere il bucket S3 quando si elimina lo stack CloudFormation.
- Applicazione della crittografia dei dati in transito
- Applica la regola del ciclo di vita per spostare le versioni degli oggetti non correnti nell'archivio Glacier dopo 90 giorni.

AWS CloudTrail

- Configurare un percorso in AWS CloudTrail per registrare gli eventi API in Amazon S3 relativi al bucket creato dal Costrutto.

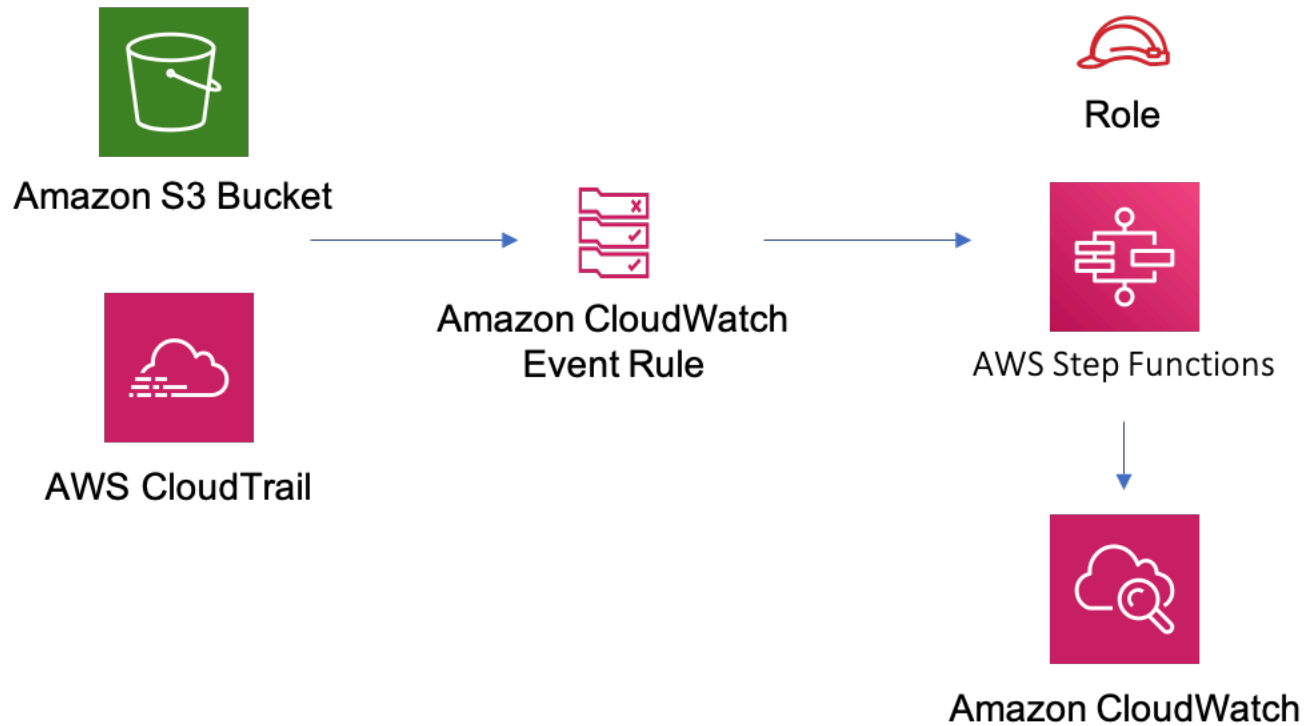
Amazon CloudWatch Events Regola

- Concedere le autorizzazioni di privilegi minimi agli eventi CloudWatch per attivare la funzione Lambda.

AWS Step Funzione

- Abilitare la registrazione CloudWatch per API Gateway.
- Distribuisci le best practice CloudWatch Allarmi per la funzione Step.

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-constructs/aws-s3-step-function](https://github.com/aws-solutions-constructs/aws-s3-step-function)

aws-sns-lambda

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Versionamento semantico](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_sns_lambda</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-sns-lambda</code> |
|  Java | <code>software.amazon.awsconstructs.services.snslambda</code> |

Overview

Questo AWS Solutions Construct implementa un Amazon SNS collegato a una funzione AWS Lambda.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id [string](#)
- props [SnsToLambdaProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------|--------------------------------------|--|
| EsistenteLambdaobj? | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFunctionProps</code> causerà un errore. |
| LambdafunctionPunps? | lambda.FunctionProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| Esistente Topicobj? | sns.Topic | Istanza esistente dell'oggetto argomento SNS, fornendo sia questo che <code>topicProps</code> causerà un errore. |
| TopicProps? | sns.TopicProps | L'utente facoltativo ha fornito proprietà per ignorare le |

| Nome | Tipo | Descrizione |
|------|------|--|
| | | proprietà predefinite per l'argomento SNS. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|----------------|---------------------------------|--|
| LambdaFunction | lambda.Function | Restituisce un'istanza della funzione Lambda creata dal pattern. |
| SNSTopic | sns.Topic | Restituisce un'istanza dell'argomento SNS creato dal modello. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

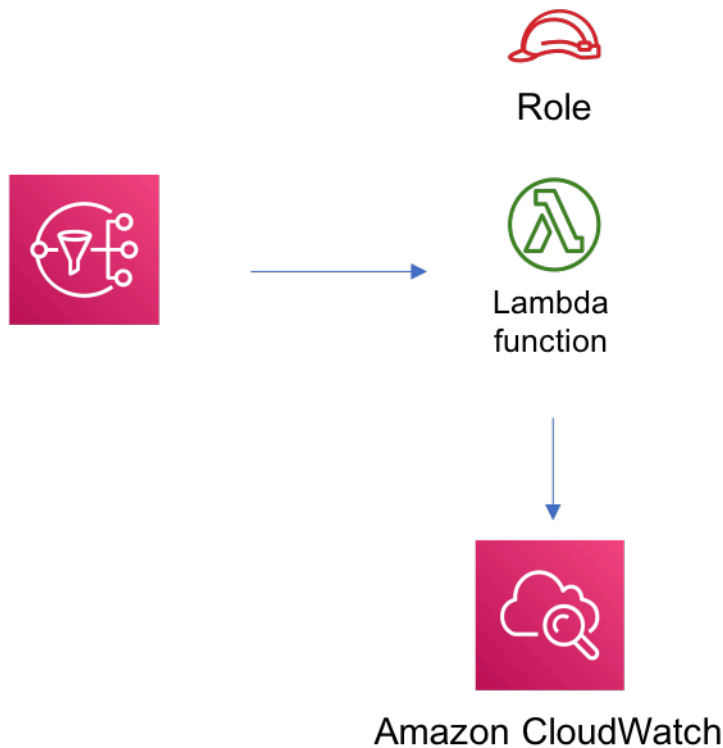
Argomento Amazon SNS

- Configurare le autorizzazioni di accesso con privilegi minimi per l'argomento SNS.
- Abilitare la crittografia lato server utilizzando la chiave AWS gestita KMS
- Applicazione della crittografia dei dati in transito

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Attivare il tracciamento X-Ray
- Impostazione delle variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:




[@aws -solutions-buils/aws-sns-lambda](https://github.com/aws-solutions-builds/aws-sns-lambda)

aws-sns-sqs

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic \(Versione semantica\) Modello](#). Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|--|
|  Python | <code>aws_solutions_constructs.aws_sns_sqs</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-sns-sqs</code> |
|  Java | <code>software.amazon.awsconstructs.services.snssqs</code> |

Overview

Questo modello AWS Solutions implementa un argomento Amazon SNS collegato a una coda Amazon SQS.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- scope [Construct](#)
- id `string`
- props [SnsToSqsProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|---------------------|--------------------------------|---|
| Esistente TopicObj? | sns.Topic | Istanza esistente dell'oggetto argomento SNS, fornendo sia questo che <code>topicProps</code> causerà un errore. |
| TopicProps? | sns.TopicProps | Proprietà facoltative fornite dall'utente per ignorare le proprietà predefinite per l'argomento SNS. Ignorato se <code>unexistingTopicObj</code> viene fornito. |
| ExistingQueueObj? | sqs.Queue | Una coda SQS esistente facoltativa da utilizzare al posto della coda predefinita. Fornire sia questo che <code>queueProps</code> causerà un errore. |
| QueueProps? | sqs.QueueProps | Proprietà facoltative fornite dall'utente per sovrascrivere |

| Nome | Tipo | Descrizione |
|--|---|--|
| | | le proprietà predefinite per la coda SQS. Ignorato se <code>unexistingQueueObj</code> viene fornito. |
| <code>DeployDeadLetterQueue?</code> | <code>boolean</code> | Se creare una coda secondari a da utilizzare come coda SDQ. L'impostazione predefinita è <code>true</code> . |
| <code>DeadLetterQueueProps?</code> | <code>sqs.QueueProps</code> | Oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per la coda delle lettere morte. Utilizzato solo se il valore <code>deployDeadLetterQueue</code> viene impostata su <code>true</code> . |
| <code>MaxReceiveCount?</code> | <code>number</code> | Il numero di volte in cui un messaggio può essere rimosso in caso contrario prima di essere spostato nella coda SDQ. L'impostazione predefinita è 15. |
| <code>EnableEncryptionWithCustomerManagedKey?</code> | <code>boolean</code> | Se utilizzare una chiave di crittografia gestita dal cliente, gestita da questa app CDK o importata. Se si importa una chiave di crittografia, questa deve essere specificata nella casella <code>encryptionKey</code> per questo costrutto. |

| Nome | Tipo | Descrizione |
|---------------------|------------------------------|---|
| encryptionKey? | kms.Key | Chiave di crittografia opzionale esistente da utilizzare al posto della chiave di crittografia predefinita. |
| EncryptionKeyProps? | kms.KeyProps | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la chiave di crittografia. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|------------------|---------------------------|---|
| SNSTopic | sns.Topic | Restituisce un'istanza dell'argomento SNS creato dal modello. |
| encryptionKey | kms.Key | Restituisce un'istanza della chiave di crittografia creata dal modello. |
| SQSqueue | sqs.Queue | Restituisce un'istanza della coda SQS creata dal pattern. |
| DeadLetterQueue? | sqs.Queue | Restituisce un'istanza della coda di lettere morte creata dal modello, se ne viene distribuita una. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

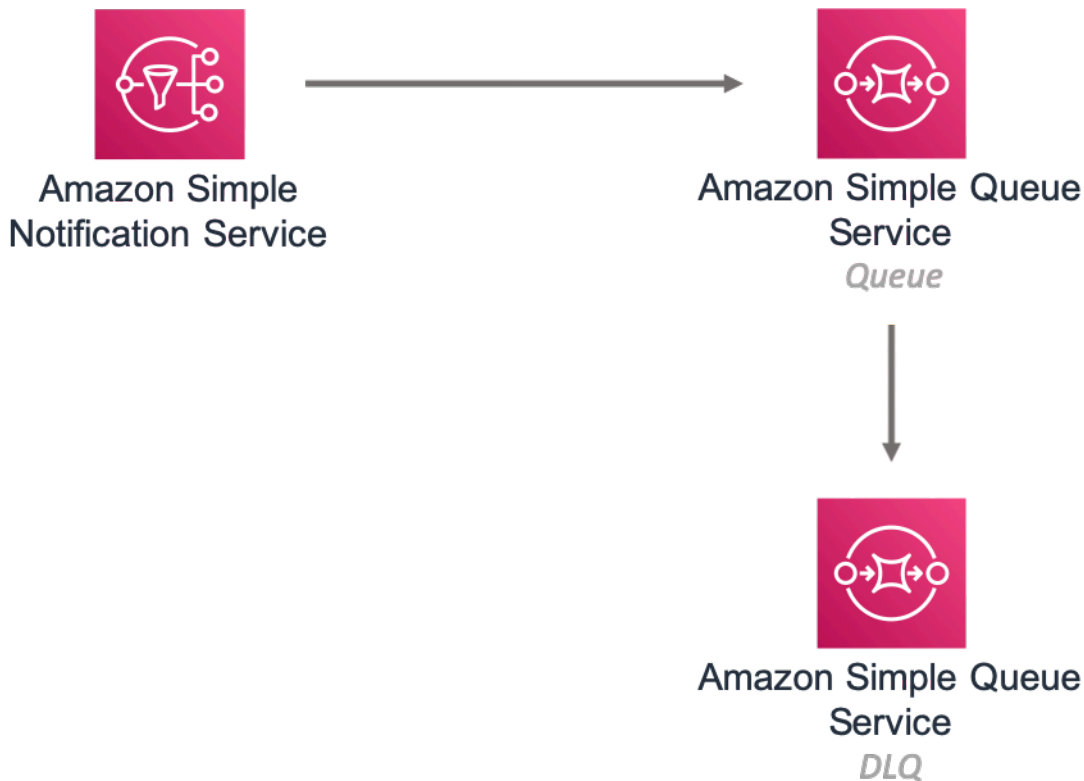
Argomento Amazon SNS

- Configurare le autorizzazioni di accesso con privilegi minimi per l'argomento SNS.
- Abilitare la crittografia lato server utilizzando la chiave AWS gestita.
- Applicazione della crittografia dei dati in transito

Coda Amazon SQS

- Configurare le autorizzazioni di accesso con privilegi minimi per la coda SQS.
- Distribuire la coda SDQ per la coda SQS di origine.
- Abilitare la crittografia lato server per la coda SQS utilizzando la chiave KMS gestita dal cliente.
- Applicazione della crittografia dei dati in transito

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-buils/aws-sns-sqs](#)

aws-sqs-lambda

STABILITY

EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic versioning](#) Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

Nota: Per garantire la corretta funzionalità, i pacchetti AWS Solutions Constructs e i pacchetti AWS CDK nel progetto devono essere della stessa versione.

| Linguaggio | Pacchetto |
|---|---|
|  Python | <code>aws_solutions_constructs.aws_sqs_lambda</code> |
|  TypeScript | <code>@aws-solutions-constructs/aws-sqs-lambda</code> |
|  Java | <code>software.amazon.awsconstructs.services.sqslambda</code> |

Overview

Questo AWS Solutions Construct implementa una coda Amazon SQS collegata a una funzione AWS Lambda.

Ecco una definizione di modello distribuibile minima in 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);
```

Parametri

- `scope` [Construct](#)
- `id` `string`
- `props` [SqsToLambdaProps](#)

Puntelli di costruzione modelli

| Nome | Tipo | Descrizione |
|----------------------------------|---------------------------------|--|
| <code>EsistenteLambdaobj?</code> | lambda.Function | Istanza esistente dell'oggetto Function Lambda, fornendo sia questo che <code>lambdaFun</code> |

| Nome | Tipo | Descrizione |
|-------------------------------------|--|--|
| | | <code>ctionProps</code> causerà un errore. |
| <code>LambdafunctionPunps?</code> | <u><code>lambda.FunctionProps</code></u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la funzione Lambda. Ignorato se <code>unexistingLambdaObj</code> viene fornito. |
| <code>ExistingQueueObj?</code> | <u><code>sqs.Queue</code></u> | Una coda SQS esistente facoltativa da utilizzare al posto della coda predefinita. Fornire sia questo <code>chequeueProps</code> causerà un errore. |
| <code>QueueProps?</code> | <u><code>sqs.QueueProps</code></u> | Proprietà facoltative fornite dall'utente per sovrascrivere le proprietà predefinite per la coda SQS. Ignorato se <code>unexistingQueueObj</code> viene fornito. |
| <code>DeployDeadLetterQueue?</code> | <code>boolean</code> | Indica se creare una coda secondaria da utilizzare come coda DLQ. L'impostazione predefinita è <code>true</code> . |

| Nome | Tipo | Descrizione |
|-----------------------|---------------------------------------|---|
| DeadLetterQueueProps? | <u>sqs.QueueProps</u> | Oggetti di scena facoltativi forniti dall'utente per sovrascrivere gli oggetti di scena predefiniti per la coda delle lettere morte. Utilizzato solo se il valore <code>deployDeadLetterQueue</code> viene impostata su <code>true</code> . |
| MaxReceiveCount? | <code>number</code> | Il numero di volte in cui un messaggio può essere rimosso senza successo prima di essere spostato nella coda DLQ. L'impostazione predefinita è 15. |

Proprietà modelli

| Nome | Tipo | Descrizione |
|------------------|--|---|
| DeadLetterQueue? | <u>sqs.Queue</u> | Restituisce un'istanza della coda di lettere morte creata dal modello, se ne viene distribuita una. |
| LambdaFunction | <u>lambda.Function</u> | Restituisce un'istanza della funzione Lambda creata dal pattern. |
| SQSqueue | <u>sqs.Queue</u> | Restituisce un'istanza della coda SQS creata dal pattern. |

Impostazioni predefinite

L'implementazione predefinita di questo modello senza sostituzioni imposterà i seguenti valori predefiniti:

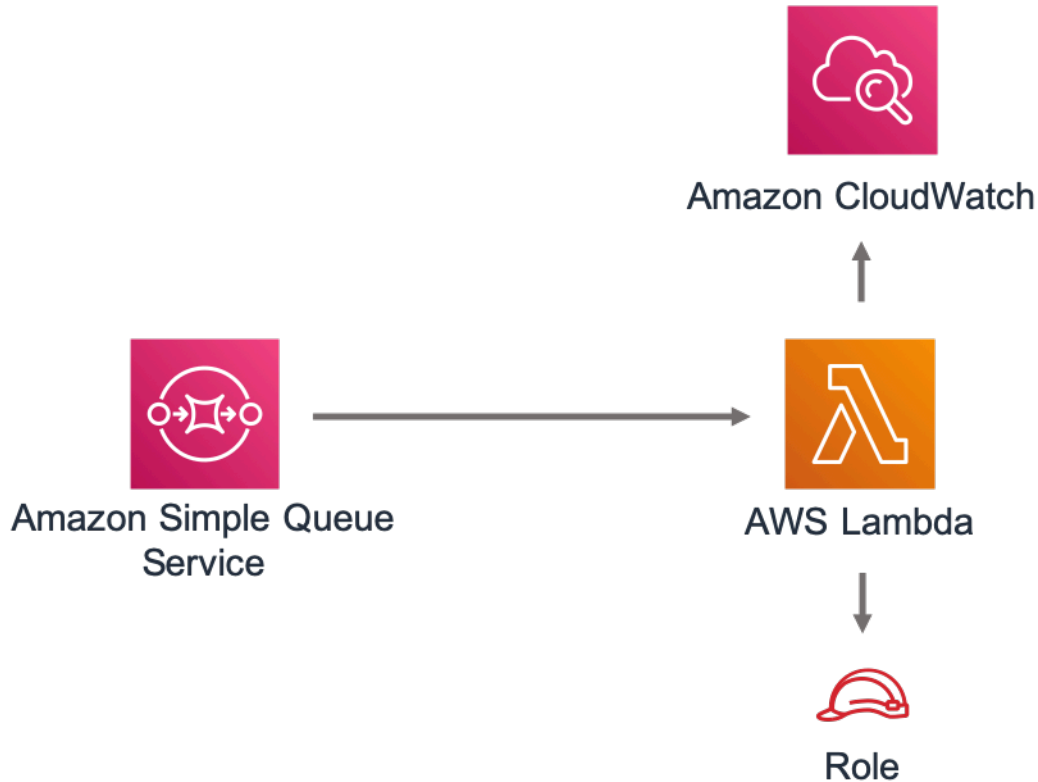
Coda Amazon SQS

- Distribuire la coda DLQ per la coda SQS di origine.
- Abilitare la crittografia lato server per la coda SQS di origine utilizzando la chiave AWS gestita.
- Applicazione della crittografia dei dati in transito

Funzione di AWS Lambda

- Configurare il ruolo IAM di accesso con privilegi limitati per la funzione Lambda.
- Abilita il riutilizzo delle connessioni con Keep-Alive per la funzione NodeJS Lambda.
- Abilitare il tracciamento X-Ray.
- Impostazione delle variabili di ambiente:
 - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(per le funzioni Node 10.x e successive)

Architecture



GitHub

Per visualizzare il codice per questo modello, creare/visualizzare problemi e richieste pull e altro ancora:



[@aws -solutions-buils/aws-sqs-lambda](#)

core

STABILITY EXPERIMENTAL

Tutte le classi sono in fase di sviluppo attivo e soggette a modifiche o rimozione non compatibili con le versioni precedenti in qualsiasi versione futura. Questi non sono soggetti alla [Semantic versioning](#) ([Controllo delle](#)Modello. Ciò significa che, mentre è possibile utilizzarli, potrebbe essere necessario aggiornare il codice sorgente quando si esegue l'aggiornamento a una versione più recente di questo pacchetto.

La libreria principale include gli elementi di base di AWS Solutions Constructs. Definisce le classi principali che vengono utilizzati nel resto di AWS Solutions Constructs.

Proprietà predefinite per i costrutti CDK AWS

La libreria di base imposta le proprietà predefinite per i costrutti CDK AWS utilizzati dai costrutti di soluzioni AWS.

Ad esempio, il seguente è lo snippet di proprietà predefinite per il costrutto S3 Bucket creato da AWS Solutions Constructs costruito. Per impostazione predefinita, attiva la crittografia lato server, il controllo delle versioni del bucket, blocca tutti gli accessi pubblici e configura la registrazione degli accessi S3.

```
{
  encryption: s3.BucketEncryption.S3_MANAGED,
  versioned: true,
  blockPublicAccess: s3.BlockPublicAccess.BLOCK_ALL,
  removalPolicy: RemovalPolicy.RETAIN,
  serverAccessLogsBucket: loggingBucket
}
```

Sostituisci le proprietà predefinite

Le proprietà predefinite impostate dalla libreria Core possono essere sostituite dalle proprietà fornite dall'utente. Ad esempio, l'utente può sostituire la proprietà Amazon S3 Block Public Access per soddisfare requisiti specifici.

```
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
  },
});
```

Avvisi della sostituzione della proprietà

Quando una proprietà predefinita della libreria Core viene sovrascritta da una proprietà fornita dall'utente, Constructs emette uno o più messaggi di avviso alla console che evidenziano le modifiche. Questi messaggi hanno lo scopo di fornire consapevolezza della situazione all'utente e prevenire sostituzioni involontarie che potrebbero creare rischi per la sicurezza. Questi messaggi verranno visualizzati ogni volta che vengono eseguiti comandi relativi alla distribuzione/compilazione, `includcdk deploy`, `cdk synth`, `npm test`, e così via.

Messaggi di esempio: `AWS_CONSTRUCTS_WARNING: An override has been provided for the property: BillingMode. Default value: 'PAY_PER_REQUEST'. You provided: 'PROVISIONED'.`

Attivazione degli avvisi di override

I messaggi di avviso di sostituzione sono abilitati per impostazione predefinita, ma possono essere attivati/disattivati esplicitamente utilizzando il comando `overrideWarningsEnabledShell` variabile.

- Per esplicitamente disattivare gli avvisi, eseguire `export overrideWarningsEnabled=false`.
- Per esplicitamente attivare gli avvisi, eseguire `export overrideWarningsEnabled=true`.
- Per ripristinare l'impostazione predefinita, eseguire `unset overrideWarningsEnabled`.

Revisioni del documento

Per ricevere notifiche sugli aggiornamenti di AWS Solutions Constructs, iscriviti al feed RSS.

| update-history-change | update-history-description | update-history-date |
|--------------------------------------|--|---------------------|
| Contenuti aggiornati | Aggiunto modello aws-lambda-ssmstringparameter. Altri aggiornamenti di contenuti minori. | 27 maggio 2021 |
| Contenuti aggiornati | Aggiunto modello aws-lambda-secretsmanager. Altri aggiornamenti di contenuti minori. | 12 maggio 2021 |
| Contenuti aggiornati | Aggiornamenti delle proprietà per selezionare i modelli *-lambda. Altri aggiornamenti di contenuti minori. | 17 aprile 2021 |
| Contenuti aggiornati | Risolto un problema nella procedura dettagliata per gli utenti Python e esempi di proprietà aggiornati per i costrutti contenenti funzioni Lambda. | 30 marzo 2021 |
| Contenuti aggiornati | Correzioni secondari/aggiornamenti agli oggetti di scena e alle impostazioni predefinite per i modelli selezionati. | 8 marzo 2021 |
| Contenuti aggiornati | Correzioni secondari/aggiornamenti al contenuto dettagliata. | 4 marzo 2021 |
| Contenuti aggiornati | Aggiuntoaws-lambda-sagemakerendpoint e le | 24 febbraio 2021 |

proprietà aggiornate per alcuni modelli Kinesis Firehose.

[Contenuti aggiornati](#)

Aggiunto `aws-kinesis-streams-gluejob` e passaggi di procedura dettagliata aggiornati per gli utenti Python. 17 febbraio 2021

[Contenuti aggiornati](#)

Proprietà aggiornate per `aws-cloudfront-*` Modelli di fantasia 9 febbraio 2021

[Contenuti aggiornati](#)

Aggiunto link a GitHub per ogni modello. 5 febbraio 2021

[Contenuti aggiornati](#)

Proprietà aggiornate per i modelli selezionati. 1 febbraio 2021

[Contenuti aggiornati](#)

Documentazione aggiornata delle proprietà e delle impostazioni predefinite per i modelli selezionati. 4 gennaio 2021

[Contenuti aggiornati](#)

Aggiunti nuovi modelli: `aws-cloudfront-mediastore` e `aws-s3-sqs`. 20 dicembre 2020

[Contenuti aggiornati](#)

Rimosso modello `aws-lambda-sagemaker`. 17 Novembre 2020

[Contenuti aggiornati](#)

Aggiunti nuovi modelli: `aws-events-rule-kinesisstream`, `aws-events-rule-kinesisfirehose-s3` e `aws-lambda-sagemaker`. 27 ottobre 2020

| | | |
|--------------------------------------|---|-------------------|
| Contenuti aggiornati | Aggiornato per riflettere il cambiamento radicale nei modelli aws-events-rule-sns e aws-events-rule-sqs: i nomi delle classi e delle interfacce sono cambiati in caso pascal. | 22 ottobre 2020 |
| Contenuti aggiornati | Aggiunti aws-apigateway-sagemakerendpoint e aws-kinesisstreams-kinesisfirehose-3 modelli; altri aggiornamenti minori ai contenuti esistenti. | 20 ottobre 2020 |
| Contenuti aggiornati | Aggiunto modello aws-apigateway-iot; altri aggiornamenti minori ai contenuti esistenti. | 7 ottobre 2020 |
| Contenuti aggiornati | Aggiornati frammenti di codice pattern distribuibili minimi e impostazioni predefinite di best practice per tutti i modelli. | 5 Ottobre 2020 |
| Contenuti aggiornati | Proprietà aggiornate per il modello aws-kinesisstreams-lambda per riflettere il cambiamento di rottura. | 14 settembre 2020 |
| Contenuti aggiornati | Correzione minore alla seconda parte della procedura dettagliata. | 10 settembre 2020 |
| Contenuti aggiornati | Aggiunti aws-apigateway-kinesisstream, aws-events-rule-sns e modelli aws-events-rule-sqs. | 10 settembre 2020 |

| | | |
|--------------------------------------|---|------------------|
| Contenuti aggiornati | Aggiunto modello aws-sns-sqs; aggiornamenti a tutti i modelli SNS; correzioni tipografiche minori. | 2 settembre 2020 |
| Contenuti aggiornati | Nomi di moduli fissi per modello aws-sqs-lambda. | 31 agosto 2020 |
| Contenuti aggiornati | Risolto il nome del modulo Python per il modello aws-dynamodb-stream-lambda-elasticsearch-kibana. | 31 agosto 2020 |
| Contenuti aggiornati | Aggiornato i valori predefiniti per i modelli Lambda; altri aggiornamenti minori. | 27 agosto 2020 |
| Contenuti aggiornati | Proprietà pubbliche aggiornate e per i modelli S3; valori predefiniti aggiornati per i modelli DynamoDB. | 10 agosto 2020 |
| Contenuti aggiornati | Aggiornato più modelli per evidenziare l'applicazione predefinita della crittografia in transito. | 4 agosto 2020 |
| Contenuti aggiornati | Aggiunto modello aws-lambda-sqs-lambda; istruzioni di configurazione migliorate nella guida introduttiva; aggiornati tutti i modelli per rendere disponibili risorse aggiuntive tramite le proprietà pubbliche. | 27 luglio 2020 |
| Contenuti aggiornati | Aggiunto modello aws-lambda-sqs; altri aggiornamenti minori. | 20 luglio 2020 |

| | | |
|--------------------------------------|--|----------------|
| Contenuti aggiornati | Rimosse le proprietà DeployBucket e DeployBucket dai modelli pertinenti; altri aggiornamenti minori. | 9 luglio 2020 |
| Contenuti aggiornati | Aggiunto modello aws-lambda-step-function e corretti errori tipografici minori. | 7 luglio 2020 |
| Contenuti aggiornati | Aggiunto TableObj? per selezionare i pattern DynamoDB. | 25 giugno 2020 |
| Contenuti aggiornati | Diverse correzioni di testo e correzioni per collegamenti interrotti. | 23 giugno 2020 |
| Versione iniziale | AWS Solutions Constructs resi pubblici. | 22 giugno 2020 |

Notices

I clienti sono responsabili di effettuare la propria valutazione indipendente delle informazioni contenute nel presente documento. Questo documento: (a) ha solo scopo informativo, (b) rappresenta le offerte e le pratiche attuali di prodotti AWS, che sono soggette a modifiche senza preavviso, e (c) non crea alcun impegno o garanzia da parte di AWS e delle sue affiliate, fornitori o licenziatari. I prodotti o i servizi AWS vengono forniti «così com'è» senza garanzie, dichiarazioni o condizioni di alcun tipo, esplicite o implicite. Le responsabilità di AWS nei confronti dei propri clienti sono definite dai contratti AWS e il presente documento non costituisce parte né modifica qualsivoglia contratto tra AWS e i suoi clienti.

© 2020 Amazon Web Services, Inc. o società affiliate. Tutti i diritti riservati.

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.