AWS Lösungen

Konstrukte für AWS S-Lösungen

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

Konstrukte für AWS S-Lösungen: AWS Lösungen

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

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Marken, die nicht Eigentum von Amazon sind, sind Eigentum ihrer jeweiligen Inhaber, die mit Amazon verbunden oder nicht verbunden oder von Amazon gesponsert oder nicht gesponsert sein können.

Table of Contents

Übersicht 1
Was ist AWS Solutions Constructs? 1
Warum AWS Solutions Constructs verwenden? 1
Erste Schritte 3
Voraussetzungen 3
Installieren des AWS CDK 4
Arbeiten mit AWS Lösungskonstruktionen 4
Exemplarische Vorgehensweise - Teil 1 5
Hello Konstrukte 5
Erstellen des App-Verzeichnisses und Initialisierung des AWS CDK 6
Aktualisieren von Projektbasis-Abhängigkeiten 7
Code für Lambda Handler 9
Installieren der Abhängigkeiten von AWS CDK und AWS Solutions Constructs 10
Hinzufügen eines Amazon API Gateway/AWS Lambda-Musters zu Ihrem Stack 12
cdk-Bereitstellen 18
Stack-Ausgaben 19
Testen Ihrer Anwendung 19
Exemplarische Vorgehensweise - Teil 2 19
Lambda Code 20
Installieren der neuen Abhängigkeiten 22
Definieren der Ressourcen 23
Überprüfen Sie die Änderungen. 36
cdk-Bereitstellen 37
Stack-Ausgabe 38
Testen Ihrer Anwendung 38
Anwendungsbeispiel für 39
Statische AWS S3-Website 40
AWS Simple serverlose Image-Handler 40
AWS Server-Web-App-Anwendungen 40
API-Referenz 42
Module 42
Module-Inhalt 42
aws-apigateway-dynamodb 43
Übersicht 43

Initialisierung 44
Muster-Requisiten 44
Muster-Eigenschaften 46
Standardeinstellungen 46
Architektur 48
GitHub 48
aws-apigateway-iot 48
Übersicht 49
Initialisierer 50
Muster-Requisiten 50
Muster-Eigenschaften 51
Standardeinstellungen 52
Architektur 55
Beispiele 55
GitHub 57
aws-apigateway-kinesisstreams 57
Übersicht 58
Initialisierer 58
Muster Konstrukt Requisiten 59
Muster-Eigenschaften 60
Beispiele zu API-Nutzung 61
Standardeinstellungen 62
Architektur 63
GitHub 63
aws-apigateway-lambda 63
Übersicht 64
Initialisierer 64
Muster-Konstrukt Requisiten 65
Muster-Eigenschaften 66
Standardeinstellungen 66
Architektur 67
GitHub 68
aws-apigateway-sagemakerendpunkt 68
Übersicht 69
Initialisierer 69
Muster-Konstrukt Requisiten 70

Muster-Eigenschaften 71
Beispiele zu -API-Verwendung 61
Standardeinstellungen 72
Architektur 73
GitHub 73
aws-apigateway-sqs 74
Übersicht 74
Initialisierer 75
Muster-Konstrukt 75
Muster-Eigenschaften 77
Beispiele zu Nutzungssproben 61
Standardeinstellungen 79
Architektur 80
GitHub 80
aws-cloudfront-apiggateway 80
Übersicht 81
Initialisierer 82
Muster-Konstrukt 82
Muster-Eigenschaften 83
Standardeinstellung 84
Architektur 84
GitHub 85
aws-cloudfront-apigateway-lambda 85
Übersicht 86
Initialisierer 86
Muster-Konstrukt Requisiten 86
Muster-Eigenschaften 88
Standardeinstellung 89
Architektur 90
GitHub 90
aws-cloudfront-mediastore 90
Übersicht 91
Initialisierer 91
Muster Konstrukt Requisiten 92
Muster-Eigenschaften 93
Standardeinstellung 94

Architektur 95
GitHub 95
aws-cloudfront-s3 95
Übersicht 96
Initialisierer 96
Muster-Konstrukt 97
Muster-Eigenschaften 98
Standardeinstellung 99
Architektur 100
GitHub 100
aws-cognito-apigateway-lambda 100
Übersicht 81
Initialisierer 102
Muster-Konstrukt 103
Muster-Eigenschaften 104
Standardeinstellungen 105
Architektur 106
GitHub 106
AWS-Dynamodb-Stream-Lambda 107
Übersicht 107
Initialisierer 108
Muster-Konstrukt 108
Muster-Eigenschaften 109
Lambda-Funktion 110
Standardeinstellungen 110
Architektur 111
GitHub 111
aws-dynamodb-stream-lambda-elasticsearch-kibana 111
Übersicht 112
Initialisierer 113
Muster-Konstrukt Requisiten 113
Muster-Eigenschaften 115
Lambda-Funktion 116
Standardeinstellungen 116
Architektur 117
GitHub 118

aws-events-regel-kinesisfireschlauch-s3 118
Übersicht 119
Initialisierer 119
Muster-Konstrukt 120
Muster-Eigenschaften 121
Standardeinstellung 122
Architektur 123
GitHub 123
aws-events-regel-kinesisstreams 123
Übersicht 124
Initialisierer 125
Muster-Konstrukt Requisiten 125
Muster-Eigenschaften 126
Standardeinstellung 126
Architektur 127
GitHub 127
aws-events-regel-Lambda 127
Übersicht 128
Initialisierer 129
Muster-Konstrukt Requisiten 129
Muster-Eigenschaften 130
Standardeinstellung 130
Architektur 131
GitHub 131
aws-events-rule-sns 131
Übersicht 132
Initialisierer 133
Muster-Konstrukt Requisiten 133
Muster-Eigenschaften 135
Standardeinstellung 135
Architektur 136
GitHub 136
aws-events-regel-sqs 136
Übersicht 137
Initialisierer 138
Muster-Konstrukt 138

Muster-Eigenschaften 141
Standardeinstellung 141
Architektur 142
GitHub 142
aws-events-regel-Schritt-Funktion 143
Übersicht 143
Initialisierer 144
Muster-Konstrukt Requisiten 144
Muster-Eigenschaften 145
Standardeinstellungen 146
Architektur 146
GitHub 147
aws-iot-kinesisfireschlauch-s3 147
Übersicht 148
Initialisierer 148
Muster-Requisiten 149
Muster-Eigenschaften 150
Standardeinstellung 151
Architektur 152
GitHub 152
aws-iot-lambda 152
Übersicht 153
Initialisierer 154
Muster-Konstrukt 154
Muster-Eigenschaften 155
Standardeinstellungen 155
Architektur 156
GitHub 156
aws-iot-lambda-dynamodb 156
Übersicht 157
Initialisierer 158
Muster-Konstrukt Requisiten 158
Muster-Eigenschaften 159
Standardeinstellungen 160
Architektur 161
GitHub 161

aws-kinesisfireschlauch-s3 161
Übersicht 162
Initialisierer 162
Muster-Konstrukt Requisiten 163
Muster-Eigenschaften 164
Standardeinstellung 164
Architektur 165
GitHub 166
aws-kinesisfireschlauch-s3-und-kinesisanalytics 166
Übersicht 167
Initialisierer 168
Muster-Konstrukt Requisiten 168
Muster-Eigenschaften 169
Standardeinstellungen 170
Architektur 171
GitHub 171
aws-kinesisstreams-gluejob 171
Übersicht 172
Initialisierer 173
Musterkonstrukt Requisiten 174
SinkDataStoreProps 176
SinkStoreType 176
Standardeinstellung 176
Architektur 178
GitHub 178
aws-kinesisstreams-kinesisfireschlauch-s3 179
Übersicht 179
Initialisierer 180
Muster-Konstrukt Requisiten 180
Muster-Eigenschaften 182
Standardeinstellung 183
Architektur 184
GitHub 184
aws-kinesisstreams-lambda 184
Übersicht 185
Initialisierer 186

Muster-Konstrukt Requisiten 186
Muster-Eigenschaften 187
Standardeinstellung 188
Architektur 189
GitHub 189
aws-lambda-dynamodb 189
Übersicht 190
Initialisierer 191
Muster-Konstrukt 191
Muster-Eigenschaften 195
Standardeinstellung 196
Architektur 197
GitHub 197
aws-lambda-elasticsearch-kibana 198
Übersicht 198
Initialisierer 199
Muster-Konstrukt Requisiten 199
Muster-Eigenschaften 201
Lambda-Funktion 202
Standardeinstellungen 202
Architektur 203
GitHub 203
aws-lambda-S3 204
Übersicht 204
Initialisierer 205
Muster-Konstrukt 205
Muster-Eigenschaften 209
Standardeinstellung 210
Architektur 211
GitHub 211
aws-lambda-ssmstringparameter 211
Übersicht 212
Initialisierer 213
Muster-Konstrukt 213
Muster-Eigenschaften 218
Standardeinstellungen 218

Architektur 219
GitHub 219
aws-lambda-sagemakerendpunkt 220
Übersicht 220
Initialisierer 221
Muster Konstrukt Requisiten 222
Muster-Eigenschaften 226
Standardeinstellungen 227
Architektur 228
GitHub 228
aws-lambda-Sekretsmanager 228
Übersicht 229
Initialisierer 230
Muster-Konstrukt Requisiten 230
Muster-Eigenschaften 233
Standardeinstellungen 234
Architektur 235
GitHub 235
aws-lambda-sns 235
Übersicht 236
Initialisierer 237
Muster-Konstrukt Requisiten 237
Muster-Eigenschaften 241
Standardeinstellung 241
Architektur 242
GitHub 242
aws-lambda-sqs 242
Übersicht 243
Initialisierer 244
Muster-Konstrukt 244
Muster-Eigenschaften 248
Standardeinstellung 249
Architektur 250
GitHub 251
aws-lambda-sqs-lambda 251
Übersicht 252

Initialisierer ... 252
Muster-Konstrukt ... 253
Muster-Eigenschaften ... 255
Standardeinstellungen ... 256
Architektur ... 257
GitHub ... 257
aws-lambda-Schritt-Funktion ... 258
Übersicht ... 258
Initialisierer ... 259
Muster-Konstrukt Requisiten ... 259
Muster-Eigenschaften ... 260
Standardeinstellungen ... 261
Architektur ... 262
GitHub ... 262
aws-s3-lambda ... 262
Übersicht ... 263
Initialisierer ... 264
Muster-Konstrukt ... 264
Muster-Eigenschaften ... 265
Standardeinstellung ... 266
Architektur ... 267
GitHub ... 267
aws-s3-sqs ... 267
Übersicht ... 268
Initialisierer ... 268
Muster-Konstrukt Requisiten ... 269
Muster-Eigenschaften ... 272
Standardeinstellung ... 272
Architektur ... 273
GitHub ... 274
aws-s3-Schritt-Funktion ... 274
Übersicht ... 275
Initialisierer ... 275
Muster-Konstrukt ... 276
Muster-Eigenschaften ... 277
Standardeinstellungen ... 278

Architektur 279
GitHub 280
aws-sns-lambda 280
Übersicht 281
Initialisierer 281
Muster-Konstrukt 281
Muster-Eigenschaften 282
Standardeinstellungen 283
Architektur 284
GitHub 284
aws-sns-sqs 284
Übersicht 285
Initialisierer 286
Muster-Konstrukt Requisiten 286
Muster-Eigenschaften 288
Standardeinstellung 289
Architektur 290
GitHub 290
aws-sqs-lambda 290
Übersicht 291
Initialisierer 291
Muster-Requisiten 292
Muster-Eigenschaften 294
Standardeinstellung 294
Architektur 295
GitHub 295
core 295
Standardeigenschaften für AWS CDK-Konstrukte 296
Überschreiben der Standardeigenschaften 296
Zurücksetzen von Eigenschaftsüberschreibungen 297
Dokumentversionen 299
Hinweise 304
..... cccv

AWS Lösungskonstrukte

Veröffentlichungsdatum: Mai 2021(Dokumentversionen)

Was ist AWS Solutions Constructs?

AWS Solutions Constructs (Constructs) ist eine Open-Source-Erweiterung desAWS Cloud Development Kit (AWS CDK), das Multi-Service-, gut gestaltete Muster für die schnelle Definition von Lösungen in Code bietet, um vorhersehbare und wiederholbare Infrastruktur zu schaffen. Das Ziel ist es, die Erfahrung für Entwickler zu beschleunigen, um Lösungen jeder Größe mit musterbasierten Definitionen für ihre Architektur zu erstellen.

Verwenden Sie die AWS Solutions Constructs, um Ihre Lösungen in einer vertrauten Programmiersprache zu definieren. Die AWS Solutions Constructs unterstützen derzeit TypeScript, JavaScript, Python und Java.

Um den vollständigen Katalog der Muster von AWS Solutions Constructs zu durchsuchen,Klicken Sie hier.

Warum AWS Solutions Constructs verwenden?

Mit der Innovationsrate von Cloud-Anbietern kann es erschreckend sein, Best Practices zu kennen und zu verstehen und sicherzustellen, dass sie in Ihrer Lösung korrekt implementiert werden. Mit Konstrukten können Sie vorgefertigte, gut gestaltete Muster und Anwendungsfälle kombinieren, die gängige Aktionen mithilfe von Clouddiensten skalierbar und sicher ausführen. Da Constructs eine Bibliothek für moderne Programmiersprachen zur Verfügung stellt, können Sie vorhandene Entwicklungsfähigkeiten und vertraute Tools auf die Aufgabe anwenden, eine gut gestaltete Cloud-Infrastruktur für Ihre Lösungen zu erstellen.

Weitere Vorteile von AWS Solutions Constructs sind:

- Es basiert auf dem Open-Source-Software-Entwicklungs-Framework für das AWS Cloud Development Kit (AWS CDK).
- Verwenden Sie Logik (if-Anweisungen, for-Schleifen usw.), wenn Sie Ihre Lösungsinfrastruktur definieren.
- Verwenden Sie objektorientierte Techniken, um ein Modell Ihres Systems zu erstellen.

- Definieren Sie Abstraktionen auf hoher Ebene, teilen Sie sie und veröffentlichen Sie sie in Ihrem Team, Unternehmen oder Ihrer Community.
- Organisieren Sie Ihre Lösungen in logischen Modulen.
- Teilen Sie Ihre Lösung und verwenden Sie sie als Bibliothek.
- Testen Sie Ihren Infrastrukturcode mit branchenüblichen Protokollen.
- Verwenden Sie Ihren vorhandenen Code-Review-Workflow.

Das Ziel von AWS Solutions Constructs ist es, die Komplexität und die Kleberlogik zu reduzieren, die bei der Integration gängiger gut gestalteter Muster erforderlich sind, um Ihre Lösungsziele in AWS zu erreichen.

Erste Schritte mit AWS Lösungskonstruktionen

In diesem Thema wird beschrieben, wie Sie AWS Cloud Development Kit (AWS CDK), AWS Solutions Constructs installieren und konfigurieren und Ihre erste AWS CDK-App mit AWS Solutions Constructs Mustern erstellen.

> **Note**
>
> AWS Solutions Constructs wird von AWS CDK-Versionen ≥ 1.46.0 unterstützt.

> **Tip**
>
> Wollen Sie tiefer graben? Versuchen Sie esWerkstattfür einen ausführlicheren Rundgang durch ein Projekt aus der realen Welt.

> **Tip**
>
> Weitere Informationen über die ersten Schritte mit dem AWS Cloud Development Kit (AWS CDK) finden Sie imAWS CDK-Entwicklerhandbuch.

Prerequisites

AWS Solutions Constructs basiert auf dem AWS CDK. Daher müssen Sie Node.js (>= 10.3.0) installieren, auch diejenigen, die in anderen Sprachen als TypeScript oder JavaScript arbeiten. Dies liegt daran, dass dieAWS CDKund AWS Solutions Constructs werden in TypeScript entwickelt und unter Node.js ausgeführt. Die Bindungen für andere unterstützte Sprachen verwenden dieses Backend und Toolset.

Sie müssen Ihre Anmeldeinformationen und eine AWS Region angeben, um die AWS CDK-CLI verwenden zu können, wie unter Angeben Ihrer Anmeldeinformationen und Region beschrieben.

Weitere Voraussetzungen hängen wie folgt von Ihrer Entwicklungssprache ab.

Sprache	Voraussetzungen
P	Python >= 3.6
T t	TypeScript >= 2.7
J	Java = 1.8

Installieren des AWS CDK

Informationen zur Installation und Konfiguration des AWS CDK finden Sie im AWS CDK Developer Guide -Installieren des AWS CDK.

Arbeiten mit AWS Lösungskonstruktionen

Der typische Workflow zum Erstellen einer neuen App bei der Arbeit mit AWS Solutions Constructs folgt dem gleichen Ansatz wie das AWS CDK.

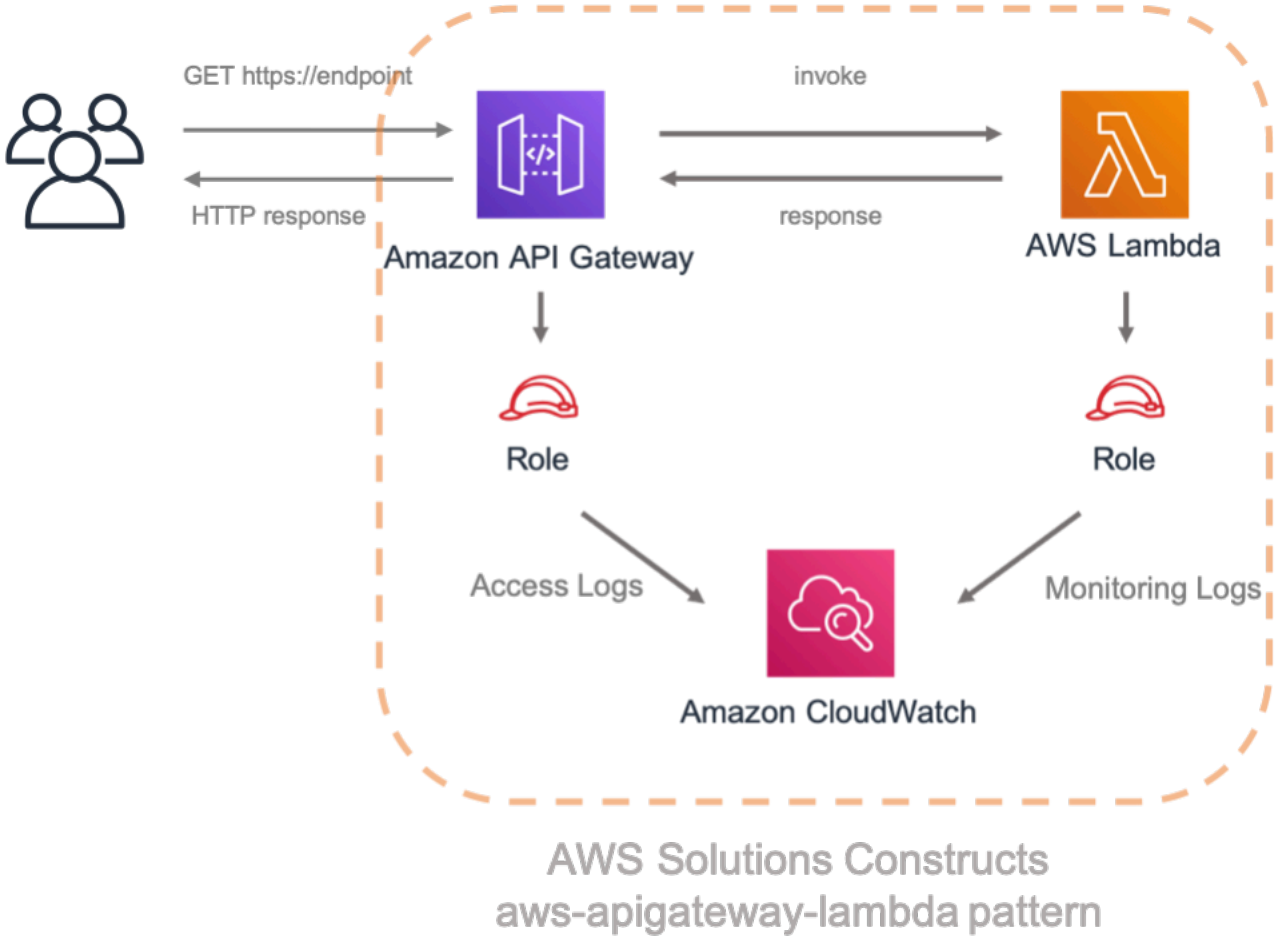
1. Erstellen Sie das App-Verzeichnis.
2. Initialisieren der Anwendung.
3. Fügen Sie die Musterabhängigkeiten von AWS Solutions Constructs hinzu.
4. Fügen Sie der App zusätzlichen Code hinzu.
5. Kompilieren Sie die App, falls erforderlich.
6. Stellen Sie die in der App definierten Ressourcen bereit.
7. Testen der Anwendung.

Wenn Probleme auftreten, durchlaufen Sie das Ändern, Kompilieren (falls erforderlich), Bereitstellen und testen Sie es erneut.

Exemplarische Vorgehensweise - Teil 1

> **Note**
>
> AWS Solutions Constructs wird in AWS CDK-Versionen ≥ 1.46.0 unterstützt.

In diesem Lernprogramm erfahren Sie, wie Sie eine einfache AWS CDK-App „Hello Constructs“ erstellen und bereitstellen, die ein Muster aus AWS Solutions Constructs verwendet, von der Initialisierung des Projekts bis zur Bereitstellung der resultierenden AWS CloudFormation Vorlage. Die Hello Constructs App erstellt die folgende einfache Lösung:



Hello Konstrukte

Beginnen wir mit der Erstellung unserer ersten AWS CDK-App mit der musterbasierten Entwicklung.

> **Note**
>
> Dies ist eine Beispieländerung von`Hello CDK!`aus[CDK-Werkstatt](). Wenn Sie zum ersten Mal das AWS CDK verwenden, empfehlen wir Ihnen, mit diesem Workshop zu beginnen, um eine praktische exemplarische Vorgehensweise zu erhalten und wie Sie das CDK beim Erstellen eines realen Projekts nutzen können.

Erstellen des App-Verzeichnisses und Initialisierung des AWS CDK

Erstellen Sie ein Verzeichnis für Ihre CDK-App, und erstellen Sie dann eine AWS CDK-App in diesem Verzeichnis.

TypeScript

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

Python

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

> **Tip**
>
> Jetzt ist ein guter Zeitpunkt, um das Projekt in Ihrer Lieblings-IDE zu öffnen und zu erkunden. Wählen Sie den entsprechenden Link aus, um mehr über die Projektstruktur zu erfahren:
>
> - [TypeScript]()
> - [Python]()

Aktualisieren von Projektbasis-Abhängigkeiten

Warning

Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen AWS Solutions Constructs und AWS CDK-Pakete dieselbe Versionsnummer in Ihrem Projekt verwenden. Wenn Sie beispielsweise AWS Solutions Constructs v.1.52.0 verwenden, müssen Sie auch AWS CDK v.1.52.0 verwenden.

Tip

Notieren Sie sich die neueste Version von AWS Solutions Constructs, und wenden Sie diese Versionsnummer auf die`VERSION_NUMBER`in den folgenden Schritten (sowohl für AWS Solutions Constructs als auch für AWS CDK-Pakete). Um alle öffentlichen Releases der Konstrukts-Bibliothek zu überprüfen,Click here.

TypeScript

Bearbeiten Sie die`package.json`-Dateienthält die folgenden Informationen:

```
"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

Bearbeiten Sie die`setup.py`-Dateienthält die folgenden Informationen:

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

Installieren Sie die Projekt-Basisabhängigkeiten.

TypeScript

```
npm install
```

Python

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

Erstellen Sie die App und führen Sie sie aus und bestätigen Sie, dass sie einen leeren Stack erstellt.

TypeScript

```
npm run build
cdk synth
```

Python

```
cdk synth
```

Sie sollten einen Stack-Code wie unten dargestellt, wobei`CDK-VERSION`ist die Version des CDK. (Ihre Ausgabe kann geringfügig von der hier gezeigten Abbildung abweichen.)

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
```

Code für Lambda Handler

Wir beginnen mit dem AWS Lambda Handler-Code.

Erstellen eines -Verzeichnisses`lambda`im Stammverzeichnis Ihres Projektbaums.

TypeScript

Fügen Sie eine Datei mit dem Namen`lambda/hello.js`-Inhalt mit folgendem Inhalt:

```
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

Fügen Sie eine Datei mit dem Namen`lambda/hello.py`-Inhalt mit folgendem Inhalt:

```
import json

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

Dies ist eine einfache Lambda Funktion, die den Text „Hallo, Constructs! Sie haben [URL-Pfad]“ getroffen. Die Ausgabe der Funktion enthält auch den HTTP-Statuscode und HTTP-Header. Diese werden von API Gateway verwendet, um die HTTP-Antwort an den Benutzer zu formulieren.

Dieses Lambda ist in JavaScript zur Verfügung gestellt. Weitere Informationen zum Schreiben von Lambda Funktionen in Ihrer Sprache finden Sie imAWS Lambda Dokumentation.

Installieren der Abhängigkeiten von AWS CDK und AWS Solutions Constructs

Die AWS Solutions Constructs werden mit einer umfangreichen Bibliothek von Konstrukten ausgeliefert. Die Bibliothek ist in Module unterteilt, eines für jedes gut gestaltete Muster. Wenn Sie beispielsweise eine Amazon API Gateway Rest-API für eine AWS Lambda Funktion definieren möchten, müssen wir die`aws-apigateway-lambda`Pattern-Bibliothek.

Außerdem müssen wir die Konstruktionsbibliothek AWS Lambda und Amazon API Gateway aus dem AWS CDK hinzufügen.

Installieren Sie das AWS Lambda Modul und alle seine Abhängigkeiten in unserem Projekt:

> **Note**
>
> Denken Sie daran, die richtige, übereinstimmende Version zu ersetzen, die sowohl für AWS Lösungskonstrukte als auch für das AWS-CDK verwendet werden soll, in das`VERSION_NUMBER`Platzhalterfelder für jeden Befehl. Fehlübereinstimmende Versionen zwischen Paketen können zu Fehlern führen.

TypeScript

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

Python

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

Als nächstes installieren Sie das Amazon API Gateway Modul und alle seine Abhängigkeiten in unserem Projekt:

TypeScript

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

Python

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

Installieren Sie schließlich die AWS Solutions Constructs`aws-apigateway-lambda`Modul und alle seine Abhängigkeiten in unser Projekt:

TypeScript

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

Python

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

Hinzufügen eines Amazon API Gateway/AWS Lambda-Musters zu Ihrem Stack

Lassen Sie uns nun das AWS Solutions Constructs-Muster für die Implementierung eines Amazon API Gateway mit einem AWS Lambda Proxy definieren.

TypeScript

Bearbeiten Sie die Datei`lib/hello-constructs.ts`-Code mit folgendem Inhalt:

```
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

Bearbeiten Sie die Dateihello_constructs/hello_constructs_stack.py-Code mit folgendem Inhalt:

```
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
                )
            )
        )
```

Das war's. Dies ist alles, was Sie tun müssen, um ein API Gateway zu definieren, das alle Anforderungen an eine AWS Lambda Funktion weiterleitet. Vergleichen wir unseren neuen Stack mit dem ursprünglichen:

TypeScript

```
npm run build
cdk diff
```

Python

```
cdk diff
```

Die Ausgabe sollte in etwa wie folgt aussehen:

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

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

```
#   #                               #        #                               #
                # S::Partition}:execute-api:${ #
#   #                               #        #                               #
                # AWS::Region}:${AWS::AccountI #
#   #                               #        #                               #
                # d}:${RestApi0C43BF4B}/test-i #
#   #                               #        #                               #
                # nvoke-stage/*/"              #
#   #                               #        #                               #
                # }                            #
#########################################################################################
# + # ${LambdaFunctionServiceRole # Allow  # sts:AssumeRole                   #
 Service:lambda.amazonaws.co #                                 #
#   # .Arn}                       #        #                               # m
                #                            #
#########################################################################################
# + # ${LambdaRestApiCloudWatchRo # Allow  # sts:AssumeRole                   #
 Service:apigateway.amazonaw #                                 #
#   # le.Arn}                     #        #                               # s.com
                #                            #
#########################################################################################
# + # arn:aws:logs:${AWS::Region} # Allow  # logs:CreateLogGroup         # AWS:
${LambdaRestApiCloudWat #                                   #
#   # :${AWS::AccountId}:*        #        # logs:CreateLogStream        # chRole}
                #                            #
#   #                               #        # logs:DescribeLogGroups      #
                #                            #
#   #                               #        # logs:DescribeLogStreams     #
                #                            #
#   #                               #        # logs:FilterLogEvents        #
                #                            #
#   #                               #        # logs:GetLogEvents           #
                #                            #
#   #                               #        # logs:PutLogEvents           #
                #                            #
#########################################################################################
# + # arn:aws:logs:${AWS::Region} # Allow  # logs:CreateLogGroup         # AWS:
${LambdaFunctionService #                                   #
#   # :${AWS::AccountId}:log-grou #        # logs:CreateLogStream        # Role}
                #                            #
#   # p:/aws/lambda/*             #        # logs:PutLogEvents           #
                #                            #
#########################################################################################
```

```
(NOTE: There may be security-related changes not in this list. See https://github.com/
aws/aws-cdk/issues/1299)

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

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

Resources
[+] AWS::Logs::LogGroup ApiGatewayToLambda/ApiAccessLogGroup
 ApiGatewayToLambdaApiAccessLogGroupE2B41502
[+] AWS::IAM::Role LambdaFunctionServiceRole LambdaFunctionServiceRole0C4CDE0B
[+] AWS::Lambda::Function LambdaFunction LambdaFunctionBF21E41F
[+] AWS::ApiGateway::RestApi RestApi RestApi0C43BF4B
```

```
[+] AWS::ApiGateway::Deployment RestApi/Deployment
 RestApiDeployment180EC503d2c6df3c8dc8b7193b98c1a0bff4e677
[+] AWS::ApiGateway::Stage RestApi/DeploymentStage.prod
 RestApiDeploymentStageprod3855DE66
[+] AWS::ApiGateway::Resource RestApi/Default/{proxy+} RestApiproxyC95856DD
[+] AWS::Lambda::Permission RestApi/Default/{proxy+}/ANY/
ApiPermission.HelloConstructsStackRestApiFDB18C2E.ANY..{proxy+}
 RestApiproxyANYApiPermissionHelloConstructsStackRestApiFDB18C2EANYproxyE43D39B3
[+] AWS::Lambda::Permission RestApi/Default/{proxy+}/ANY/
ApiPermission.Test.HelloConstructsStackRestApiFDB18C2E.ANY..{proxy+}
 RestApiproxyANYApiPermissionTestHelloConstructsStackRestApiFDB18C2EANYproxy0B23CDC7
[+] AWS::ApiGateway::Method RestApi/Default/{proxy+}/ANY RestApiproxyANY1786B242
[+] AWS::Lambda::Permission RestApi/Default/ANY/
ApiPermission.HelloConstructsStackRestApiFDB18C2E.ANY..
 RestApiANYApiPermissionHelloConstructsStackRestApiFDB18C2EANY5684C1E6
[+] AWS::Lambda::Permission RestApi/Default/ANY/
ApiPermission.Test.HelloConstructsStackRestApiFDB18C2E.ANY..
 RestApiANYApiPermissionTestHelloConstructsStackRestApiFDB18C2EANY81DBDF56
[+] AWS::ApiGateway::Method RestApi/Default/ANY RestApiANYA7C1DC94
[+] AWS::ApiGateway::UsagePlan RestApi/UsagePlan RestApiUsagePlan6E1C537A
[+] AWS::Logs::LogGroup ApiAccessLogGroup ApiAccessLogGroupCEA70788
[+] AWS::IAM::Role LambdaRestApiCloudWatchRole LambdaRestApiCloudWatchRoleF339D4E6
[+] AWS::ApiGateway::Account LambdaRestApiAccount LambdaRestApiAccount

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

Das ist schön. In diesem einfachen Beispiel mit einem gut gestalteten Muster aus den AWS Solutions Constructs wurden Ihrem Stack 21 neue Ressourcen hinzugefügt.

cdk-Bereitstellen

> **Tip**
>
> Bevor Sie Ihre erste AWS CDK-App mit einer Lambda Funktion bereitstellen können, müssen Sie Ihre AWS-Umgebung booten. Dadurch wird ein Staging-Bucket erstellt, den das AWS CDK zum Bereitstellen von Stacks mit Assets verwendet. Wenn Sie zum ersten Mal das AWS CDK für die Bereitstellung von Assets verwenden, müssen Sie die`cdk bootstrap`, um den CDK-Toolkit-Stack in Ihrer AWS Umgebung bereitzustellen.

Ok, bereit zur Bereitstellung?

```
cdk deploy
```

Stack-Ausgaben

Wenn die Bereitstellung abgeschlossen ist, werden Sie folgende Zeile bemerken:

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

Dies ist eine Stapelausgabe, die automatisch vom AWS Solutions Constructs-Muster hinzugefügt wird und die URL des API Gateway Endpunkts enthält.

Testen Ihrer Anwendung

Lassen Sie uns versuchen, diesen Endpunkt mit`curl`. Kopieren Sie die URL und führen Sie sie aus (Ihr Präfix und Ihre Region werden wahrscheinlich anders sein).

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

Die Ausgabe sollte wie folgt aussehen:

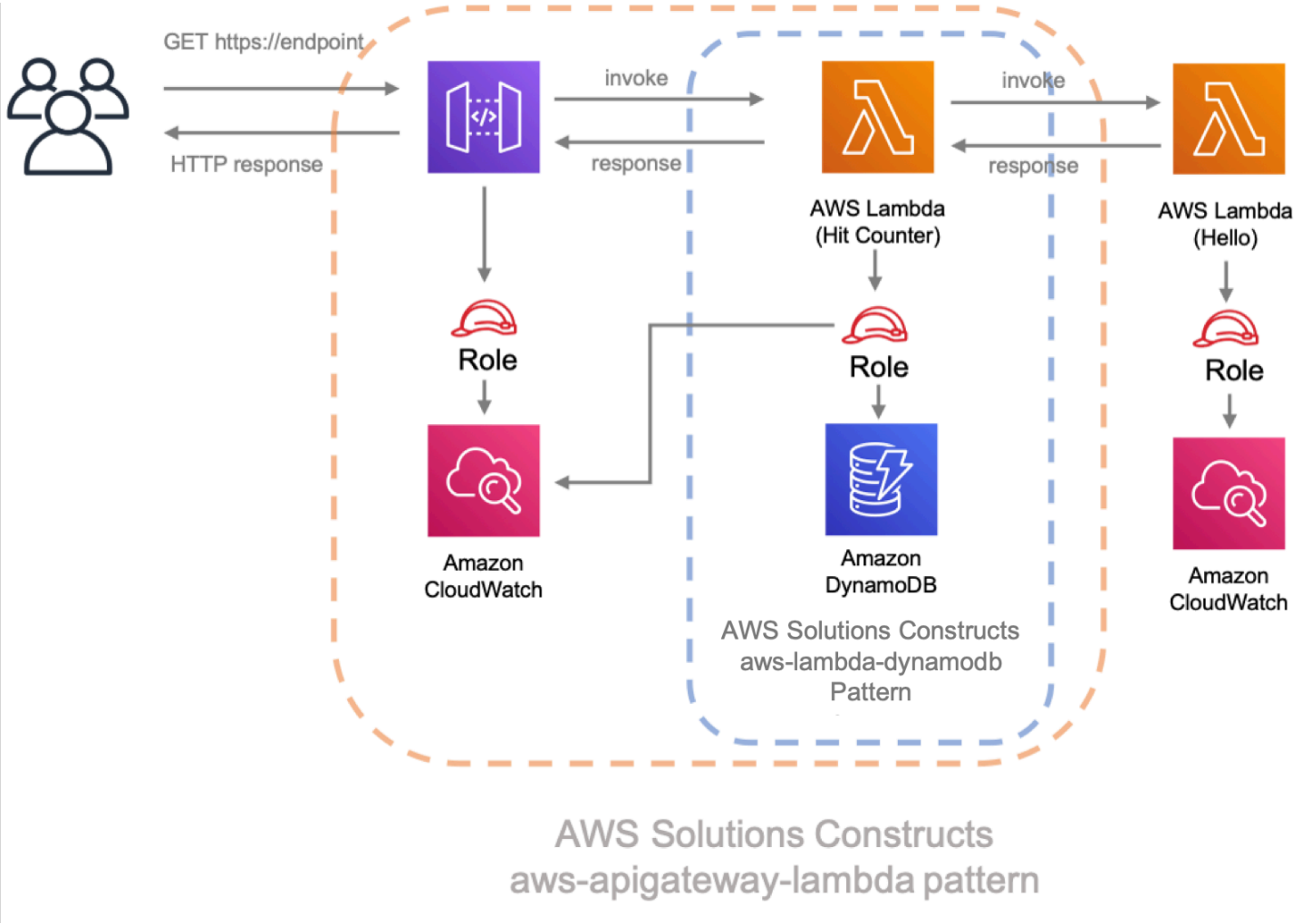
```
Hello, AWS Solutions Constructs! You've hit /
```

Wenn dies die Ausgabe ist, die Sie erhalten haben, funktioniert Ihre App!

Exemplarische Vorgehensweise - Teil 2

> **Note**
>
> AWS Solutions Constructs wird in AWS CDK-Versionen ≥ 1.46.0 unterstützt.

In diesem Tutorial erfahren Sie, wie Sie die App „Hello Constructs“ ändern, die in Teil 1. Unsere Änderung fügt einen Site-Trefferzähler hinzu, der das AWS Lambda zu DynamoDB Muster von AWS Solutions Constructs verwendet. Das Ändern der Hello Constructs App führt zu der folgenden Lösung:



Lambda Code

Beginnen wir mit dem Schreiben des Codes für die Funktion Hit Counter AWS Lambda. Diese Funktion wird Folgendes:

- Inkrementieren Sie einen Leistungsindikator, der sich auf den API-Pfad in einer Amazon DynamoDB -Tabelle bezieht,
- die Downstream-Funktion „Hello AWS Lambda“ aufrufen
- und geben Sie die Antwort an den Endbenutzer zurück.

TypeScript

Hinzufügen einer Datei`lambda/hitcounter.js`-Angabe mit folgendem Inhalt:

```
  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

Hinzufügen einer Datei`lambda/hitcounter.py`-Angabe mit folgendem Inhalt:

```
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)
```

Installieren der neuen Abhängigkeiten

> **Note**
>
> Denken Sie daran, die richtige, übereinstimmende Version zu ersetzen, die sowohl für AWS Lösungskonstrukte als auch für das AWS-CDK verwendet werden soll, in das`VERSION_NUMBER`Platzhalterfelder für jeden Befehl. Dies sollte identisch mit der Versionsnummer sein, die für Abhängigkeiten im ersten Teil dieser exemplarischen Vorgehensweise verwendet wird. Fehlübereinstimmende Versionen zwischen Paketen können zu Fehlern führen.

Wie üblich müssen wir zuerst die Abhängigkeiten installieren, die wir für unser Lösungsupdate benötigen. Zuerst müssen wir die DynamoDB Konstruktionsbibliothek installieren:

TypeScript

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

Python

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

Installieren Sie schließlich die AWS Solutions Constructs`aws-lambda-dynamodb`Modul und alle seine Abhängigkeiten in unser Projekt:

TypeScript

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

Python

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

Definieren der Ressourcen

Aktualisieren wir nun unseren Stack-Code, um unsere neue Architektur zu berücksichtigen.

Zuerst werden wir unsere neuen Abhängigkeiten importieren und die Funktion „Hallo“ außerhalb des`aws-apigateway-lambda`Muster, das wir in Teil 1 erstellt haben.

TypeScript

Bearbeiten Sie die Datei`lib/hello-constructs.ts`Folgendes:

```
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);
  }
}
```

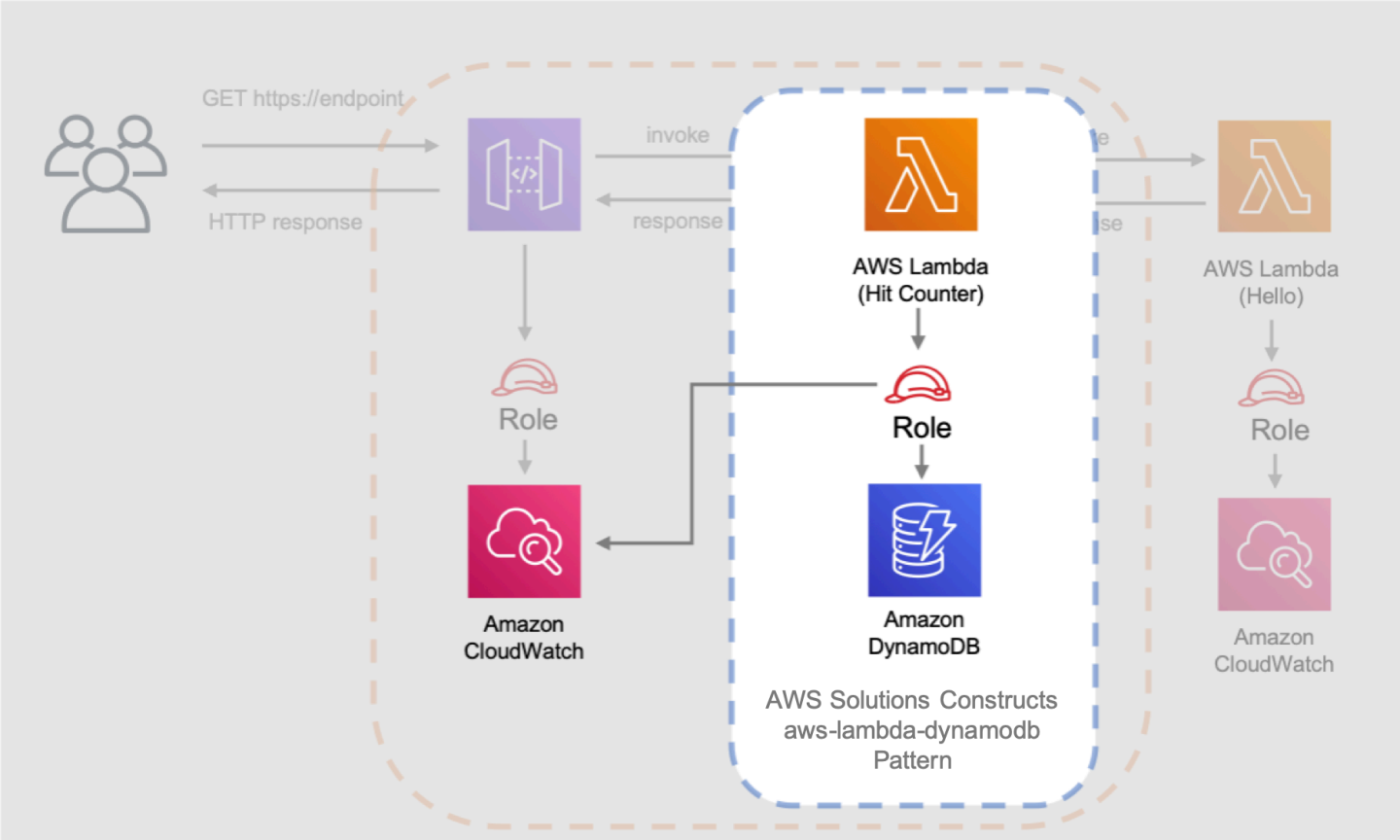
Python

Bearbeiten Sie die Dateihello_constructs/hello_constructs_stack.pyFolgendes:

```
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
                )
            )
        )
```

Als nächstes fügen wir die`aws-lambda-dynamodb`-Muster, um den Hitzähler-Service für unsere aktualisierte Architektur aufzubauen.



AWS Solutions Constructs
aws-apigateway-lambda pattern

Das nächste Update unten definiert die Eigenschaften für die`aws-lambda-dynamodb`-Muster, indem Sie die AWS Lambda Funktion mit dem Hit Counter-Handler definieren. Darüber hinaus ist die Amazon DynamoDB -Tabelle mit dem Namen`Hits`und einen Partitionsschlüssel von`path`.

TypeScript

Bearbeiten Sie die Datei`lib/hello-constructs.ts`Folgendes:

```
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

Bearbeiten Sie die Dateihello_constructs/hello_constructs_stack.pyFolgendes:

```
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
                )
            )
        )
```

Als Nächstes müssen wir die Trefferzähler-Funktion gewähren, die aus dem`aws-lambda-dynamodb`Muster hinzugefügt oben Berechtigung, unsere Hello Funktion aufzurufen.

TypeScript

Bearbeiten Sie die Datei`lib/hello-constructs.ts`Folgendes:

```
  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

Bearbeiten Sie die Dateihello_constructs/hello_constructs_stack.pyFolgendes:

```
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
                )
            )
        )
```

Schließlich müssen wir unsere ursprünglicheaws-apigateway-lambda-Muster verwenden, um unsere neue Trefferzähler-Funktion zu verwenden, die mit deraws-lambda-dynamodbMuster oben.

TypeScript

Bearbeiten Sie die Dateilib/hello-constructs.tsFolgendes:

```
  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

Bearbeiten Sie die Dateihello_constructs/hello_constructs_stack.pyFolgendes:

```
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
                )
            )
        )
```

Überprüfen Sie die Änderungen.

Lassen Sie uns unser Projekt erstellen und die Änderungen an unseren Ressourcen überprüfen, die passieren, wenn wir dies bereitstellen:

```
npm run build
cdk diff
```

Unsere Ausgabe sollte wie folgt aussehen:

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

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

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

cdk-Bereitstellen

Okay, bereit für die Bereitstellung?

```
cdk deploy
```

Stack-Ausgabe

Wenn die Bereitstellung abgeschlossen ist, werden Sie folgende Zeile bemerken:

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

Testen Ihrer Anwendung

Lassen Sie uns versuchen, diesen Endpunkt mit curl zu treffen. Kopieren Sie die URL und führen Sie sie aus (Ihr Präfix und Ihre Region werden wahrscheinlich anders sein).

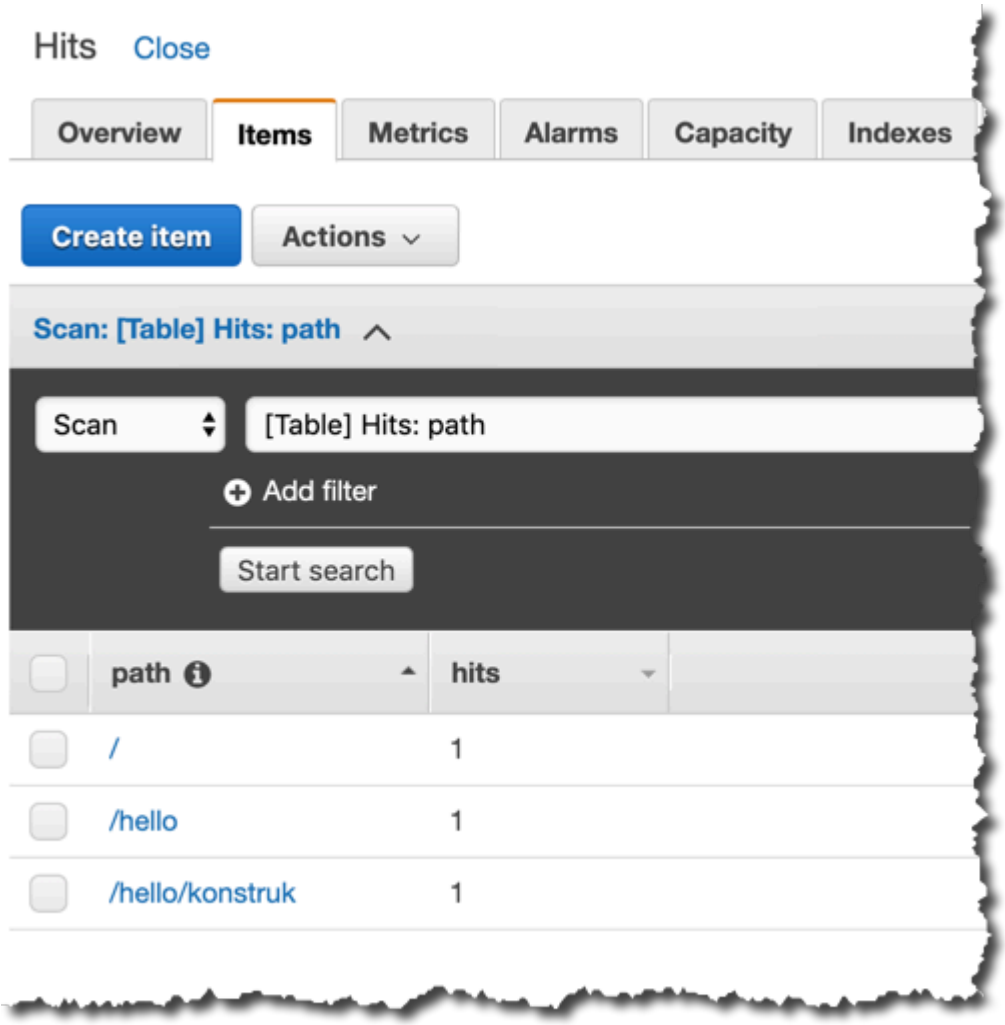
```
curl https://xxxxxxxxxx.execute-api.us-east-1.amazonaws.com/prod/
```

Die Ausgabe sollte wie folgt aussehen:

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

Rufen Sie jetzt die`Hits`Amazon DynamoDB -Tabelle.

1. Rufen Sie die DynamoDB Konsole auf.
2. Stellen Sie sicher, dass Sie sich in der Region befinden, in der Sie die Tabelle erstellt haben.
3. SelectTabellenKlicken Sie im Navigationsbereich auf und wählen Sie die OptionTreffer-Tabelle.
4. Öffnen Sie die Tabelle und wählen Sie „Items“.
5. Sie sollten sehen, wie viele Treffer Sie für jeden Pfad haben.



6. Versuchen Sie, einen neuen Pfad zu treffen, und aktualisieren Sie die Ansicht Elemente. Sie sollten ein neues Element mit einem`hits`Anzahl von eins.

Wenn dies die Ausgabe ist, die Sie erhalten haben, funktioniert Ihre App!

Anwendungsbeispiel für

Diese Bibliothek enthält eine Sammlung funktionaler Anwendungsfallimplementierungen, um die Verwendung von Konstrukts Architekturmustern zu demonstrieren. Diese können auf die gleiche Weise wie architektonische Muster verwendet werden und können als zusätzliche „höhere“ Abstraktion dieser Muster konzeptualisiert werden. Die folgenden Anwendungsfälle werden als Funktionsbeispiele bereitgestellt:

Statische AWS S3-Website

Dieses Anwendungsfallmuster (`aws-s3-static-website`) implementiert eine Amazon CloudFront Distribution, einen Amazon S3 Bucket und eine AWS Lambda-basierte benutzerdefinierte Ressource, um den statischen Websiteinhalt für die Wild Rydes-Demo-Website zu kopieren (Teil des`aws-serverless-web-app`-Implementierung).

Quellcode (aws-s3-static-website)

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

AWS Simple serverlose Image-Handler

Dieses Anwendungsfallmuster (`aws-serverless-image-handler`) implementiert eine Amazon CloudFront Distribution, eine Amazon API Gateway REST-API, eine AWS Lambda Funktion und die erforderlichen Berechtigungen/Logik, um eine funktionale Image-Handler-API für die Bereitstellung von Image-Inhalten aus einem oder mehreren Amazon S3 Buckets innerhalb des Bereitstellungskontos bereitzustellen.

Quellcode (aws-serverless-image-handler)

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

AWS Server-Web-App-Anwendungen

Dieses Anwendungsfallmuster (`aws-serverless-web-app`) implementiert eine einfache serverlose Webanwendung, die es Benutzern ermöglicht, Einhorn-Fahrten von der Wild Rydes-Flotte anzufordern. Die Anwendung stellt Benutzern eine HTML-basierte Benutzeroberfläche zur Angabe des Standorts vor, an dem sie abgeholt werden möchten, und wird auf dem Backend mit einem RESTful-Web-Service Schnittstelle, um die Anfrage einzureichen und ein nahegelegenes Einhorn zu versenden. Die Anwendung bietet auch Möglichkeiten für Benutzer, um sich beim Service zu registrieren und sich vor dem Anfordern von Fahrten anzumelden.

Quellcode (aws-serverless-web-app)

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

API-Referenz

AWS Solutions Constructs (Constructs) ist eine Open-Source-Erweiterung des AWS Cloud Development Kit (AWS CDK), die mit mehreren Diensten gut konstruierte Muster zur schnellen Definition von Lösungen in Code zur Erstellung vorhersehbarer und wiederholbarer Infrastruktur bereitstellt. Das Ziel von Constructs ist es, die Erfahrung für Entwickler zu beschleunigen, um Lösungen jeder Größe mit musterbasierten Definitionen für ihre Architektur zu erstellen.

Bei den in Konstrukten definierten Mustern handelt es sich um hochrangige Multi-Service-Abstraktionen von AWS CDK-Konstrukten mit Standardkonfigurationen, die auf gut konzipierten Best Practices basieren. Die Bibliothek ist in logische Module mit objektorientierten Techniken organisiert, um jedes architektonische Mustermodell zu erstellen.

Das CDK ist in den folgenden Sprachen verfügbar:

- JavaScript, TypeScript (Node.js ≥ 10.3.0)
- Python (Python ≥ 3,6)
- Java (Java ≥ 1,8)

Modules

AWS Solutions Constructs ist in mehrere Module gegliedert. Sie sind wie folgt benannt:

- aws-xxx: Gut gestaltetes Musterpaket für die angegebenen Leistungen. Dieses Paket enthält Konstrukte, die mehrere AWS CDK-Service-Module enthalten, um das angegebene Muster zu konfigurieren.
- xxx: Pakete, die nicht starten“- At...„sind Konstrukt-Kernmodule, die verwendet werden, um Best Practice-Standardwerte für Dienste zu konfigurieren, die in der Pattern-Bibliothek verwendet werden.

Module-Inhalt

Diese Module enthalten die folgenden Arten:

- Muster- Alle übergeordneten Multi-Services-Konstrukte in dieser Bibliothek.
- Andere Typen- Alle nicht-konstruierten Klassen, Schnittstellen, Strukturen und Enums, die existieren, um die Muster zu unterstützen.

Muster nehmen eine Reihe von (Eingabe-) Eigenschaften in ihrem Konstruktor; der Satz von Eigenschaften (und welche benötigt werden) kann auf der Dokumentationsseite eines Musters angezeigt werden.

Die Dokumentationsseite des Musters listet auch die verfügbaren Methoden auf, die aufgerufen werden sollen, und die Eigenschaften, die verwendet werden können, um Informationen über das Muster abzurufen, nachdem es instanziiert wurde.

aws-apigateway-dynamodb

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

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

Overview

Dieses AWS Solutions Construct implementiert eine Amazon API Gateway REST-API, die mit einer Amazon DynamoDB -Tabelle verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`ApiGatewayToDynamoDBProps`

Muster-Requisiten

Name	Typ	Beschreibung
DynamoTableProps	`dynamodb.TableProps`	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für DynamoDB - Tabelle
ApigatewayProps?	`api.RestApiProps`	Optionale, vom Benutzer bereitgestellte Requisite n zum Überschreiben der Standard-Requisiten für das API Gateway.

Name	Typ	Beschreibung
AllowCreateOperation	`boolean`	Gibt an, ob API Gateway -Methode für den Create-Vorgang in der DynamoDB -Tabelle bereitgestellt werden
CreateRequestTemplate	`string`	API Gateway -Anforderungsvorlage für die Create-Methode, erforderlich, wenn AllowCreateOperation auf true gesetzt ist
AllowreadOperation	`boolean`	Gibt an, ob APAPI Gateway -Methode für den Lesevorgang in der DynamoDB -Tabelle
AllowUpdateOperation	`boolean`	Gibt an, ob die API Gateway -Methode für den Update-Vorgang in der DynamoDB -Tabelle
UpdateRequestTemplate	`string`	API Gateway -Anforderungsvorlage für die Update-Methode, erforderlich, wenn AllowUpdateOperation auf true festgelegt ist
AllowDeleteOperation	`boolean`	Gibt an, ob die API Gateway -Methode für den Löschvorgang in der DynamoDB -Tabelle
LogGroupProps?	`logs.LogGroupProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardprops für die CloudWatch Logs-Protokollgruppe.

Muster-Eigenschaften

Name	Typ	Beschreibung
ApiGateway-	`api.RestApi`	Gibt eine Instanz der API Gateway REST-API zurück, die durch das Muster erstellt wurde.
ApigatewayCloudWatchRole	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster erstellt wurde, das die Zugriffsprotokollierung von der API Gateway REST-API zu CloudWatch ermöglicht.
ApigatewayLogGroup	`logs.LogGroup`	Gibt eine Instanz der Protokoll gruppe zurück, die durch das Muster erstellt wurde, an das API Gateway REST-API-Zugriffsprotokolle gesendet werden.
ApigatewayRole	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster für die API Gateway REST-API erstellt wurde.
DynamoTable	`dynamodb.Table`	Gibt eine Instanz der DynamoDB -Tabelle zurück, die durch das Muster erstellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

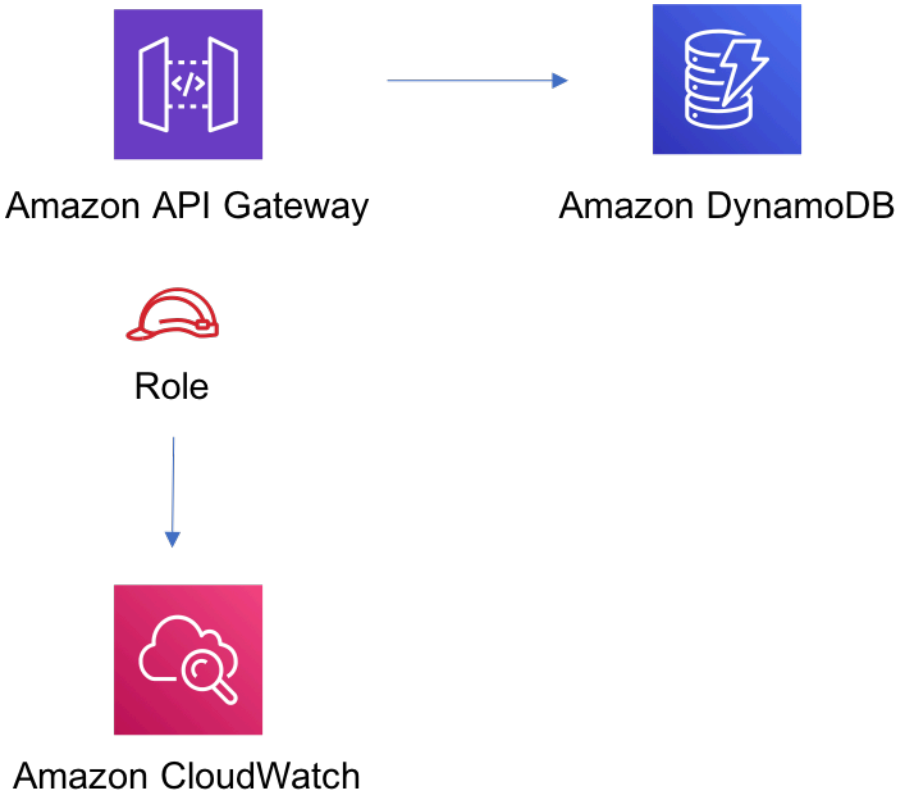
Amazon API Gateway

- Bereitstellen eines Edge-optimierten API-Endpunkts
- Aktivieren der CloudWatch Protokollierung für API Gateway
- Konfigurieren der IAM-Rolle für den Zugriff auf die geringste Berechtigung für API Gateway
- Legen Sie den standardmäßigen AuthorizationType für alle API-Methoden auf IAM fest
- Aktivieren der X-Ray blaufverfolgung

Amazon DynamoDB Tabelle

- Festlegen des Abrechnungsmodus für DynamoDB -Tabelle auf On-Demand (Bezahlung pro Anforderung)
- Aktivieren der serverseitigen Verschlüsselung für DynamoDB Table mit AWS verwaltetem KMS-Schlüssel
- Erstellt einen Partitionsschlüssel namens 'id' für DynamoDB -Tabelle
- Beibehalten der Tabelle beim Löschen des CloudFormation -Stacks
- Ermöglicht kontinuierliche Sicherungen und zeitpunktbezogene Wiederherstellung

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-konstrukte/aws-apigateway-dynamodb

aws-apigateway-iot

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_apigateway_iot`
TypeScript	`@aws-solutions-constructs/aws-apigateway-iot`
Java	`software.amazon.awsconstructs.services.apigatewayiot`

Overview

Dieses AWS Solutions Construct implementiert eine Amazon API Gateway REST-API, die mit dem AWS IoT Muster verbunden ist.

Dieses Konstrukt erstellt einen skalierbaren HTTPS-Proxy zwischen API Gateway und AWS IoT. Dies ist praktisch, wenn Legacy-Geräte, die das MQTT- oder MQTT/WebSocket-Protokoll nicht unterstützen, mit der AWS IoT Plattform interagieren möchten.

Diese Implementierung ermöglicht die Veröffentlichung von schreibgeschützten Nachrichten zu bestimmten MQTT-Themen und unterstützt außerdem Schattenaktualisierungen von HTTPS-Geräten für zulässige Elemente in der Geräteregistrierung. Es umfasst keine Lambda Funktionen zum Proxygen von Nachrichten und stützt sich stattdessen auf die direkte Integration von API Gateway zu AWS IoT, die sowohl JSON-Nachrichten als auch binäre Nachrichten unterstützt.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`ApiGatewayToIotProps`

Muster-Requisiten

Name	Typ	Beschreibung
IOtendPoint	`string`	Die AWS IoT Endpunkt-Subdomain, in die das API Gateway integriert werden soll (z. B. a1234567890123-ats).
ApigatewayCreateApikey?	`boolean`	Wenn auf gesetzt`true`wird ein API-Schlüssel erstellt und einem UsagePlan zugeordnet. Der Benutzer sollte den Header `x-api-key` angeben, während er auf RestApi zugreift. Standardwert auf`false`.
ApigatewayExecutionRole?	`iam.Role`	Die IAM-Rolle, die von API Gateway für den Zugriff auf AWS IoT verwendet wird. Wenn nicht angegeben, wird eine Standardrolle mit

Name	Typ	Beschreibung
		Platzhalterzeichen ('*') auf alle Themen und Dinge erstellt.
ApigatewayProps?	api.restApiProps	Optionale, vom Benutzer bereitgestellte Requisite n zum Überschreiben der Standard-Requisiten für die API Gateway REST API.
LogGroupProps?	logs.LogGroupProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardp rops für die CloudWatch Logs s-Protokollgruppe

Muster-Eigenschaften

Name	Typ	Beschreibung
ApiGateways	api.RestApi	Gibt eine Instanz der API Gateway REST-API zurück, die durch das Muster erstellt wurde.
ApigatewayCloudWatchRole	iam.Role	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster erstellt wurde, das die Zugriffsprotokollierung von der API Gateway REST-API zu CloudWatch ermöglicht.
ApigatewayLogGroup	logs.LogGroup	Gibt eine Instanz der Protokoll gruppe zurück, die durch das Muster erstellt wurde, an das API Gateway REST-API-

Name	Typ	Beschreibung
		Zugriffsprotokolle gesendet werden.
ApigatewayRolle	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster für die API Gateway REST-API erstellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon API Gateway

- Bereitstellen eines Edge-optimierten API-Endpunkts
- Erstellt API-Ressourcen mit`POST`Veröffentlichen von Nachrichten in IoT -Themen
- Erstellt API-Ressourcen mit`POST`Methode zum Veröffentlichen von Nachrichten`ThingShadow`und`NamedShadows`
- Aktivieren von CloudWatch Protokollierung für API Gateway
- Konfigurieren der IAM-Rolle für API Gateway mit Zugriff auf alle Themen und Dinge
- Legen Sie den standardmäßigen AuthorizationType für alle API-Methoden auf IAM fest.
- X-Ray Nachverfolgung aktivieren
- Erstellt einen UsagePlan und ordnet`prod`stage

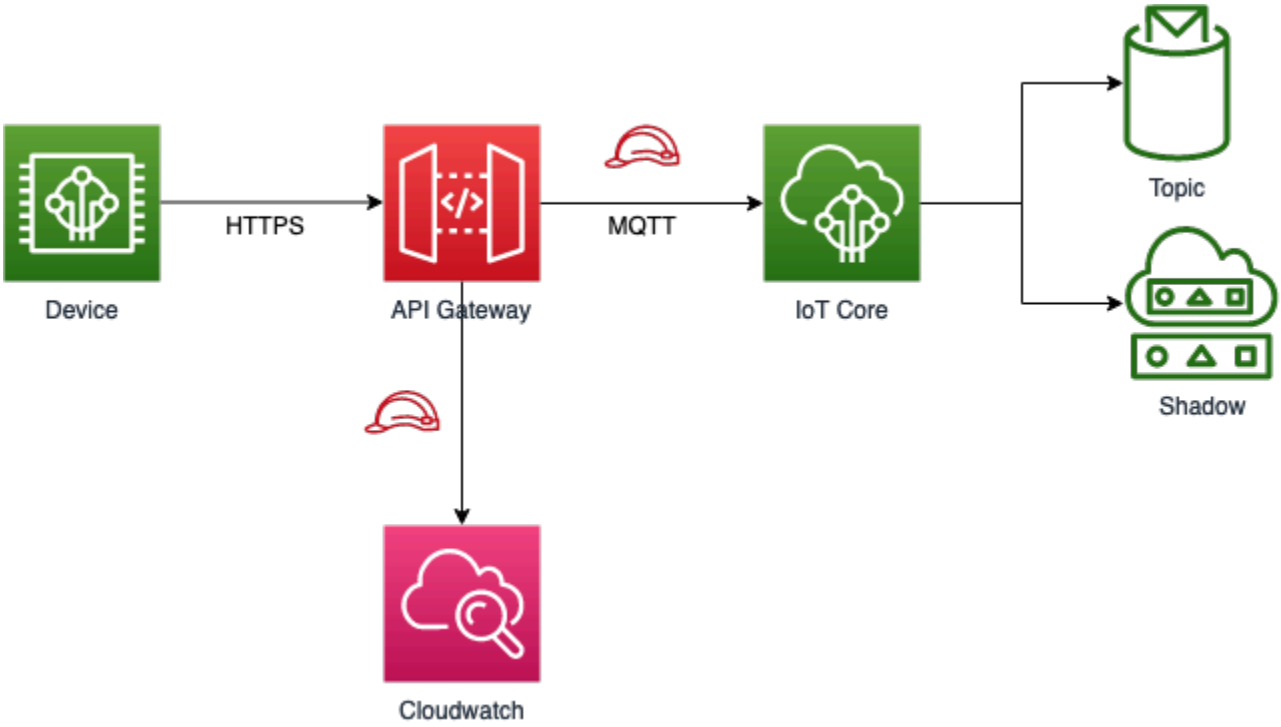
Im Folgenden finden Sie eine Beschreibung der verschiedenen Ressourcen und Methoden, die das API Gateway nach der Bereitstellung des Construct zur Verfügung stellt. Siehe dieBeispielefinden Sie weitere Informationen zum einfachen Testen dieser Endpunkte mithilfe von`curl`.

Art	Ressource	Abfragepa rameter (e)	Rückgabecode (e)	Beschreibung
`POST`	`/message/ <topics>`	`qos`	`200/403/500`	Wenn Sie diesen Endpunkt aufrufen,

Art	Ressource	Abfragepa rameter (e)	Rückgabecode (e)	Beschreibung
				müssen Sie die Themen weiterlei ten, die Sie veröffentlichen möchten (z.B. `/message/ device/fo o `).
POST	/shadow/< thingName>	Keine	200/403/500	Diese Route ermöglicht es, das Schattend okument einer Sache zu aktualisieren, da seinethingName Unbenannt er (klassisc her) Schatten verwenden. Die Karosserie muss der Standards chattenstuktur entsprechen, die einestateKnoten und zugeordne tendesiredundreporte Knoten. Siehe dieAktualisieren von Gerätesch atten-Abschnitt für ein Beispiel.

Art	Ressource	Abfragepa rameter (e)	Rückgabecode (e)	Beschreibung
POST	`/shadow/<thingName>/<shadowName>`	Keine	`200/403/500`	Diese Route ermöglicht es, das benannte Schattend okument einer Sache zu aktualisieren, da seine `thingName` und die `shadowNam e` mit dem Typ „Benannter Schatten“. Die Karosserie muss der Standards chattenstuktur entsprechen, die eine `state`Knoten und zugeordne ten `desired` und `reporte` Knoten. Siehe die Aktualisieren von benannten Schatten-Abschnitt für ein Beispiel.

Architecture



Examples

Die folgenden Beispiele funktionieren nur bei`API_KEY`, da die IAM-Autorisierung erfordert, dass auch ein SigV4-Token angegeben werden muss, stellen Sie sicher, dass die`apiGatewayCreateApiKey`-Eigenschaft Ihrer Construct Requisiten auf`true`während der Bereitstellung des Stacks, sonst funktionieren die folgenden Beispiele nicht.

Veröffentlichen einer Nachricht

Sie können mithilfe von`curl`, um eine Nachricht zu verschiedenen MQTT-Themen mithilfe der HTTPS-API zu veröffentlichen. Das folgende Beispiel wird eine Nachricht auf der`device/foo`-Thema.

```
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"}'
```

Hinweis: Ersetzen Sie die`stage-id`,`region`, und`api-key`-Parameter mit Ihren Bereitstellungswerten.

Sie können Themennamen in der URL verketten und die API akzeptiert bis zu 7 Unterthemen, auf denen Sie veröffentlichen können. Im folgenden Beispiel veröffentlicht eine Nachricht zum Themadevice/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"}'
```

Aktualisieren von Geräteschatten

Um das Schattendokument zu aktualisieren, das einer bestimmten Sache zugeordnet ist, können Sie eine Schattenstatusanforderung mit einem Sachnamen ausstellen. Siehe das folgende Beispiel zum Aktualisieren eines Thing-Shadow.

```
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" }}}'
```

Aktualisieren von benannten Schatten

Um das Schattendokument zu aktualisieren, das mit dem Namen „Schatten“ eines bestimmten Dings verknüpft ist, können Sie eine Schattenstatusanforderung mit einem Ding- und Schattennamen ausgeben. Im folgenden Beispiel erfahren Sie, wie Sie einen benannten Schatten aktualisieren.

```
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" }}}'
```

Senden binärer Nutzlasten

Es ist möglich, eine binäre Nutzlast an die Proxy-API bis zum AWS IoT -Service zu senden. Im folgenden Beispiel senden wir den Inhalt desREADME.mdDatei, die diesem Modul zugeordnet ist (als binäre Daten behandelt), umdevice/foo-Thema mithilfe derapplication/octet-streamContent-Typ.

```
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
```

Hinweis: Führen Sie diesen Befehl aus, während Sie sich im Verzeichnis dieses Projekts befinden. Sie können dann testen, andere Arten von Binärdateien aus Ihrem Dateisystem zu senden.

GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:	
	@aws -solutions-konstrukte/aws-apigateway-iot

aws-apigateway-kinesisstreams

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_apigateway_kinesisstreams`
	`@aws-solutions-constructs/aws-apigateway-kinesisstreams`

Sprache	Paket
Typoskript	
Java	`software.amazon.awsconstructs.services.apigatewaykinesisstreams`

Overview

Dieses Muster implementiert eine Amazon API Gateway REST-API, die mit einem Amazon Kinesis Datenstrom verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`ApiGatewayToKinesisStreamsProps`

Muster Konstrukt Requisiten

Name	Typ	Beschreibung
ApigatewayProps?	`api.RestApiProps`	Optionale, vom Benutzer bereitgestellte Requisite n zum Überschreiben der Standard-Requisiten für die API Gateway REST-API
PutRecordRequestTemplate?	`string`	API Gateway -Anforder ungsvorlage für die PutRecord -Aktion Wenn nicht angegeben , wird ein Standardwert verwendet.
PutRecordRequestModel?	`api.ModelOptions`	API Gateway -Anforder ungsmodell für die PutRecord -Aktion Wenn nicht angegeben , wird ein Standardwert erstellt.
PutRecordsRequestTemplate?	`string`	API Gateway Anforderu ngsvorlage für die PutRecord s -Aktion. Wenn nicht angegeben, wird ein Standardwert verwendet.
PutRecordRequestModel?	`api.ModelOptions`	API Gateway -Anforder ungsmodell für die PutRecords Aktion Wenn nicht angegeben, wird ein Standardwert erstellt.
Bestehend StreamObj?	`kinesis.Stream`	Vorhandene Instanz von Kinesis Stream, die sowohl diese als auch`kinesisSt`

Name	Typ	Beschreibung
		`reamProps` führt zu einem - Fehler.
KinesisStreamProps?	`kinesis.StreamProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für den Kinesis Stream.
LogGroupProps?	`logs.LogGroupProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardp rops für die CloudWatch Logs s-Protokollgruppe

Muster-Eigenschaften

Name	Typ	Beschreibung
Apigateway-Version	`api.RestApi`	Gibt eine Instanz der API Gateway REST-API zurück, die durch das Muster erstellt wurde.
ApigatewayRole	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster für die API Gateway REST-API erstellt wurde.
ApigatewayCloudWatchRole	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster erstellt wurde, das die Zugriffsprotokollierung von der API Gateway REST-API zu CloudWatch ermöglicht.

Name	Typ	Beschreibung
ApigatewayLogGroup	logs.LogGroup	Gibt eine Instanz der Protokoll gruppe zurück, die durch das Muster erstellt wurde, an das API Gateway REST-API-Zugriffsprotokolle gesendet werden.
KinesisStream	kinesis.Stream	Gibt eine Instanz des Kinesis Streams zurück, der durch das Muster erstellt wurde.

Beispiele zu API-Nutzung

Art	Anfordern	Anforderungstext	Warteschlangen-Aktion	Beschreibung
POST	/record	{ "data": "Hello World!", "partitionKey": "pk001" }	kinesis:PutRecord	Schreibt einen einzelnen Datensatz in den Stream.
POST	/records	{ "records": [{ "data": "abc", "partitionKey": "pk001"	kinesis:PutRecords	Schreibt mehrere Datensätze in einem einzigen Aufruf in den Stream.

Art	Anfordern	Anforderungstext	Warteschlangen-Aktion	Beschreibung

		<pre>},
		
		{ "data":
		"xyz",
		
		"partitio
		nKey":
		"pk001"
		}
		]
		}</pre>		

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

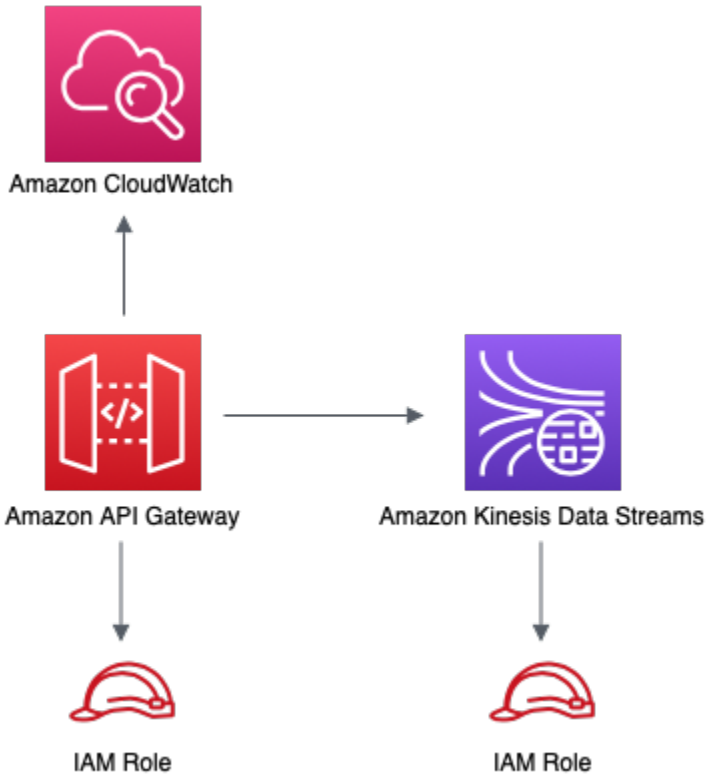
Amazon API Gateway

- Stellen Sie einen Edge-optimierten API-Endpunkt bereit.
- Aktivieren Sie die CloudWatch Protokollierung für API Gateway.
- Konfigurieren Sie die IAM-Rolle für den Zugriff auf die geringste Berechtigung für API Gateway.
- Legen Sie den standardmäßigen AuthorizationType für alle API-Methoden auf IAM fest.
- Aktivieren der X-Ray blaufverfolgung.
- Überprüfen Sie den Anforderungstext, bevor Sie Daten an Kinesis übergeben.

Amazon Kinesis Data Stream

- Konfigurieren Sie IAM-Rolle für den Zugriff auf die geringste Berechtigung für Kinesis Stream.
- Aktivieren Sie die serverseitige Verschlüsselung für Kinesis Stream mit AWS Managed KMS Key.

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-constructs/aws-apigateway-kinesisstreams

aws-apigateway-lambda

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_apigateway_lambda`
TypoScript	`@aws-solutions-constructs/aws-apigateway-lambda`
Java	`software.amazon.awsconstructs.services.apigatewaylambda`

Overview

Dieses AWS Solutions Construct implementiert eine Amazon API Gateway REST-API, die mit einer AWS Lambda Funktion verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsApiGatewayToLambdaProps

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
Gibt es LambdaObj?	lambda.Function	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auchlambdaFunctionProps führt zu einem -Fehler.
LambdaFunctionProps?	lambda.FunctionProps	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn einexistingLambdaObj wird bereitgestellt.
ApigatewayProps?	api.LambdaRestApiProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für die API.
LogGroupProps?	logs.LogGroupProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardprops für die CloudWatch Logs-Protokollgruppe.

Muster-Eigenschaften

Name	Typ	Beschreibung
ApigatewayCloudWatchRole	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster erstellt wurde, das die Zugriffsprotokollierung von der API Gateway REST-API zu CloudWatch ermöglicht.
ApigatewayLogGroup	`logs.LogGroup`	Gibt eine Instanz der Protokoll gruppe zurück, die durch das Muster erstellt wurde, an das API Gateway REST-API-Zugriffsprotokolle gesendet werden.
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
ApiGateway-Software	`api.LambdaRestApi`	Gibt eine Instanz der API Gateway REST-API zurück, die durch das Muster erstellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon API Gateway

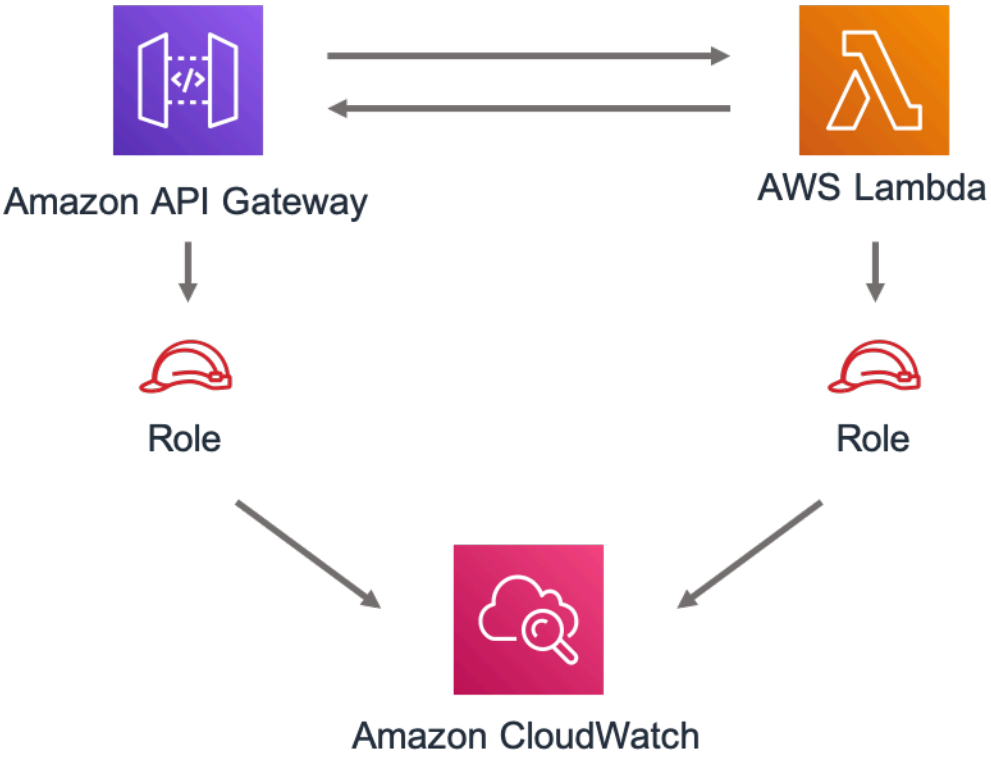
- Bereitstellen eines Edge-optimierten API-Endpunkts
- Aktivieren der CloudWatch Protokollierung für API Gateway
- Konfigurieren der IAM-Rolle für den Zugriff auf die geringste Berechtigung für API Gateway

- Legen Sie den standardmäßigen AuthorizationType für alle API-Methoden auf IAM fest.
- X-Ray blaufverfolgung aktivieren
- Festlegen von Umgebungsvariablen:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

AWS Lambda-Funktion

- Konfigurieren der IAM-Rolle für den eingeschränkten Berechtigungszugriff für die Lambda - Funktion
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion
- X-Ray blaufverfolgung aktivieren

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:	
	@aws -solutions-konstrukte/aws-apigateway-lambda

aws-apigateway-sagemakerendpunkt

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantic VersioningModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_apigateway_sagemakerendpoint`
Typoskript	`@aws-solutions-constructs/aws-apigateway-sagemakerendpoint`
Java	`software.amazon.awsconstructs.services.apigatewaysagemakerendpoint`

Overview

Dieses AWS Solutions Construct implementiert eine Amazon API Gateway REST-API, die mit einem Amazon SageMaker Endpunkt verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsApiGatewayToSageMakerEndpointProps

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
ApigatewayProps?	api.RestApiProps	Optionale, vom Benutzer bereitgestellte Requisite n zum Überschreiben der Standard-Requisiten für die API Gateway REST-API
ApigatewayExecutionRole?	iam.Role	IAM-Rolle, die vom API Gateway zum Aufrufen des SageMaker Endpoints verwendet wird. Wenn nicht angegeben, wird eine Standardrolle mit Zugriff aufendpointName .
EndpointName	string	Name des bereitgestellten SageMaker Inferenz-Endpunkts.
ResourceName?	string	Optionaler Ressourcenname, bei dem die GET-Methode verfügbar ist.
resourcePath	string	Ressourcenpfad für die GET-Methode. Die hier definierte Variable kann inrequestMa ppingTemplate .
RequestMappingTemplate	string	Mapping-Vorlage zum Konvertieren von GET-Anfor

Name	Typ	Beschreibung
		derungen, die über die REST-API empfangen wurden, in POST-Anfragen, die vom SageMaker End
ResponseMappingTemplate?	`string`	Optionale Mapping-V orlage zum Konvertieren von Antworten, die vom SageMaker Endpunkt empfangen wurden.
LogGroupProps?	`logs.LogGroupProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardp rops für die CloudWatch Logs s-Protokollgruppe.

Muster-Eigenschaften

Name	Typ	Beschreibung
ApiGateway	`api.LambdaRestApi`	Gibt eine Instanz der API Gateway REST-API zurück, die durch das Muster erstellt wurde.
ApigatewayRole	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster für die API Gateway REST-API erstellt wurde.
ApigatewayCloudWatchRole	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster erstellt wurde, das die Zugriffsprotokollierung von der

Name	Typ	Beschreibung
		API Gateway REST-API zu CloudWatch ermöglicht.
ApigatewayLogGroup	`logs.LogGroup`	Gibt eine Instanz der Protokoll gruppe zurück, die durch das Muster erstellt wurde, an das API Gateway REST-API-Zugriffsprotokolle gesendet werden.

Beispiele zu -API-Verwendung

Hinweis: Jeder SageMaker Endpunkt ist eindeutig, und die Antwort der API hängt vom bereitgestellten Modell ab. Das folgende Beispiel geht davon aus, dass die Probe vonBlogbeitrag. Eine Referenz dazu, wie das implementiert wird, finden Sie unterinteg.apigateway-sagemakerendpoint-overwrite.ts.

Art	Anfrage-Pfad	Abfrageze ichenfolge	SageMaker Aktion	Beschreibung
GET	`/321`	`items=101,131,162`	`sagemaker:InvokeEndpoint`	Ruft die Vorhersagen für einen bestimmten Benutzer und Elemente ab.

Standardeinstellungen

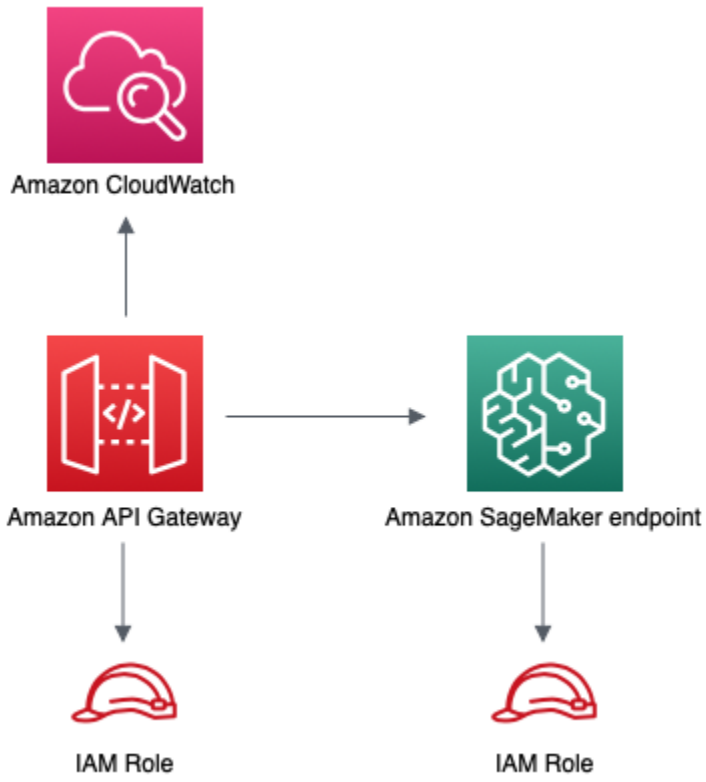
Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon API Gateway

- Bereitstellen eines Edge-optimierten API-Endpunkts
- Aktivieren der CloudWatch Protokollierung für API Gateway

- Konfigurieren der IAM-Rolle für den Zugriff auf die geringste Berechtigung für API Gateway
- Legen Sie den standardmäßigen AuthorizationType für alle API-Methoden auf IAM fest
- X-Ray blaufverfolgung
- Überprüfen Sie Anforderungsparameter, bevor Sie Daten an SageMaker

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-konstrukte/aws-apigateway-sagemakerendpoint

aws-apigateway-sqs

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_apigateway_sqs`
TypeScript	`@aws-solutions-constructs/aws-apigateway-sqs`
Java	`software.amazon.awsconstructs.services.apigatewaysqs`

Overview

Dieses AWS Solutions Construct implementiert eine Amazon API Gateway REST-API, die mit einer Amazon SQS Warteschlange verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsApiGatewayToSqsProps

Muster-Konstrukt

Name	Typ	Beschreibung
ApigatewayProps?	api.RestApiProps	Optionale, vom Benutzer bereitgestellte Requisite n zum Überschreiben der Standard-Requisiten für das API Gateway.
QueueProps?	sqs.QueueProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für die Warteschl ange.
DeployDeadletterQueue?	boolean	Gibt an, ob eine sekundäre Warteschlange bereitgestellt wird, die als Warteschlange für unzustellbare Nachricht en dient. Standardeinstellung: true.
MaxReceiveCount	number	— Die Anzahl, wie oft eine Nachricht nicht erfolgreich

Name	Typ	Beschreibung
		entfernt werden kann, bevor sie in die Warteschlange für unzustellbare Nachrichten verschoben wird.
AllowCreateOperation?	`boolean`	Gibt an, ob eine API Gateway -Methode für Create-Operationen in der Warteschlange bereitgestellt werden soll (dh SQS:SendMessage).
CreateRequestTemplate?	`string`	Überschreiben Sie die standardmäßige API Gateway -Anforderungsvorlage für die Create-Methode, wenn `allowCreateOperation` wird auf gesetzt `true`.
ReadOperation zulassen?	`boolean`	Gibt an, ob eine API Gateway -Methode für Lesevorgänge in der Warteschlange bereitgestellt werden soll (dh SQS: ReceiveMessage).
ReadRequestTemplate?	`string`	Überschreiben Sie die standardmäßige API Gateway -Anforderungsvorlage für die Read-Methode, wenn `allowReadOperation` wird auf gesetzt `true`.

Name	Typ	Beschreibung
AllowDeleteOperation?	`boolean`	Gibt an, ob eine API Gateway -Methode für Löschvorgänge in der Warteschlange bereitgestellt werden soll (dh SQS:DeleteMessage).
DeleteRequestTemplate?	`string`	Überschreiben Sie die standardmäßige API Gateway -Anforderungsvorlage für die Delete-Methode, wenn `allowDeleteOperation` wird auf gesetzt `true`.
LogGroupProps?	`logs.LogGroupProps`	Optionale, die vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardprops für die CloudWatch Logs-Protokollgruppe

Muster-Eigenschaften

Name	Typ	Beschreibung
ApiGateway-	`api.RestApi`	Gibt eine Instanz der API Gateway REST-API zurück, die durch das Muster erstellt wurde.
ApigatewayCloudWatchRole	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster erstellt wurde, das die Zugriffsprotokollierung von der

Name	Typ	Beschreibung
		API Gateway REST-API zu CloudWatch ermöglicht.
ApigatewayLogGroup	`logs.LogGroup`	Gibt eine Instanz der Protokoll gruppe zurück, die durch das Muster erstellt wurde, an das API Gateway REST-API-Zugriffsprotokolle gesendet werden.
ApigatewayRole	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster für die API Gateway REST-API erstellt wurde.
DeadLetterQueue?	`sqs.Queue`	Gibt eine Instanz der Warteschlange für toten Buchstaben zurück, die durch das Muster erstellt wurde, wenn eine bereitgestellt wird.
SQSQueue	`sqs.Queue`	Gibt eine Instanz der SQS-Warteschlange zurück, die durch das Muster erstellt wurde.

Beispiele zu Nutzungssproben

Art	Anfrage-Pfad	Anforderungstext	Warteschlangen-Aktion	Beschreibung
GET	`/`		`sqs::ReceiveMessage`	Ruft eine Nachricht aus der Warteschlange

Art	Anfrage-Pfad	Anforderungstext	Warteschlangen-Aktion	Beschreibung
POST	`/`	`{ "data": "Hello World!" }`	`sqs::SendMessage`	Übergibt eine Nachricht an die Warteschlange
DELETE	`/message?receiptHandle=[value]`		`sqs::DeleteMessage`	Löscht eine angegebene Nachricht aus der Warteschlange

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

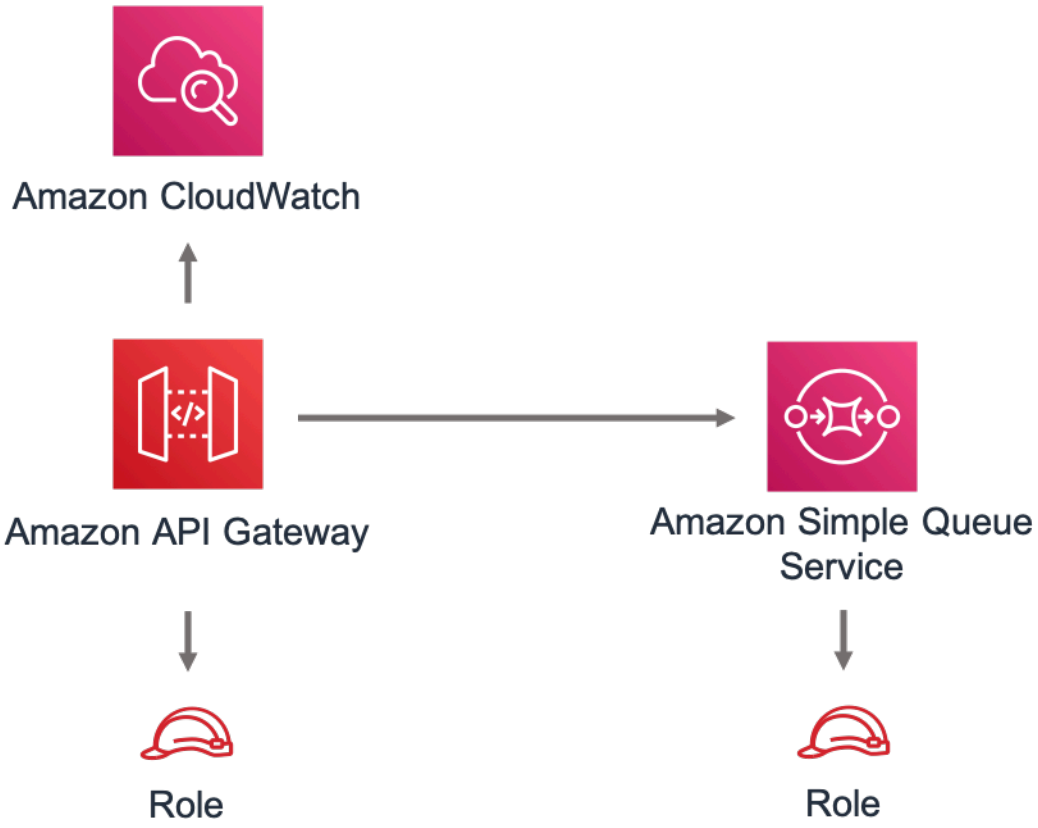
Amazon API Gateway

- Bereitstellen eines Edge-optimierten API-Endpunkts
- Aktivieren der CloudWatch Protokollierung für API Gateway
- Konfigurieren der IAM-Rolle für den Zugriff auf die geringste Berechtigung für API Gateway
- Legen Sie den standardmäßigen AuthorizationType für alle API-Methoden auf IAM fest
- Aktivieren der X-Ray blaufverfolgung

Amazon SQS-Warteschlange

- Bereitstellen einer SQS-Warteschlange für unzustellbare Nachrichten für die SQS-Quellwarteschlange
- Aktivieren der serverseitigen Verschlüsselung für Quell-SQS-Warteschlange mithilfe von AWS verwalteten KMS-Schlüssel
- Erzwingen der Verschlüsselung von Daten während der Übertragung

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-apigateway-sqs

aws-cloudfront-apiggateway

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht denSemantic version (Semantic version)Modell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

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

Overview

Dieses AWS Solutions Construct implementiert eine Amazon CloudFront Distribution vor einer Amazon API Gateway REST-API.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsCloudFrontToApiGatewayProps

Muster-Konstrukt

Name	Typ	Beschreibung
ExistingApigatewayobj	api.RestApi	Das regionale API Gateway, das mit dem CloudFront
CloudFrontDistributionProps?	cloudfront.DistributionProps	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für

Name	Typ	Beschreibung
		die CloudFront Distribution zu überschreiben.
InsertHTTPSecurityHeader?	`boolean`	Optional vom Benutzer bereitgestellte Requisiten zum Ein- und Ausschalten der automatischen Einfügung von Best Practice-HTTP-Sicherheitsheadern in allen Antworten von CloudFront

Muster-Eigenschaften

Name	Typ	Beschreibung
Apigateway-Version	`api.RestApi`	Gibt eine Instanz der API Gateway REST-API zurück, die durch das Muster erstellt wurde.
CloudFrontLoggingBucket?	`s3.Bucket`	Gibt eine Instanz des Protokollierungsbuckets zurück, der durch das Muster für die CloudFront Webverteilung erstellt wurde.
CloudFrontWebDistribution	`cloudfront.CloudFrontWebDistribution`	Gibt eine Instanz der CloudFront Webverteilung zurück, die durch das Muster erstellt wurde.
EdgeLambdaFunctionVersion?	`lambda.Version`	Gibt eine Instanz der Lambda Kantenfunktionsversion zurück, die durch das Muster erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

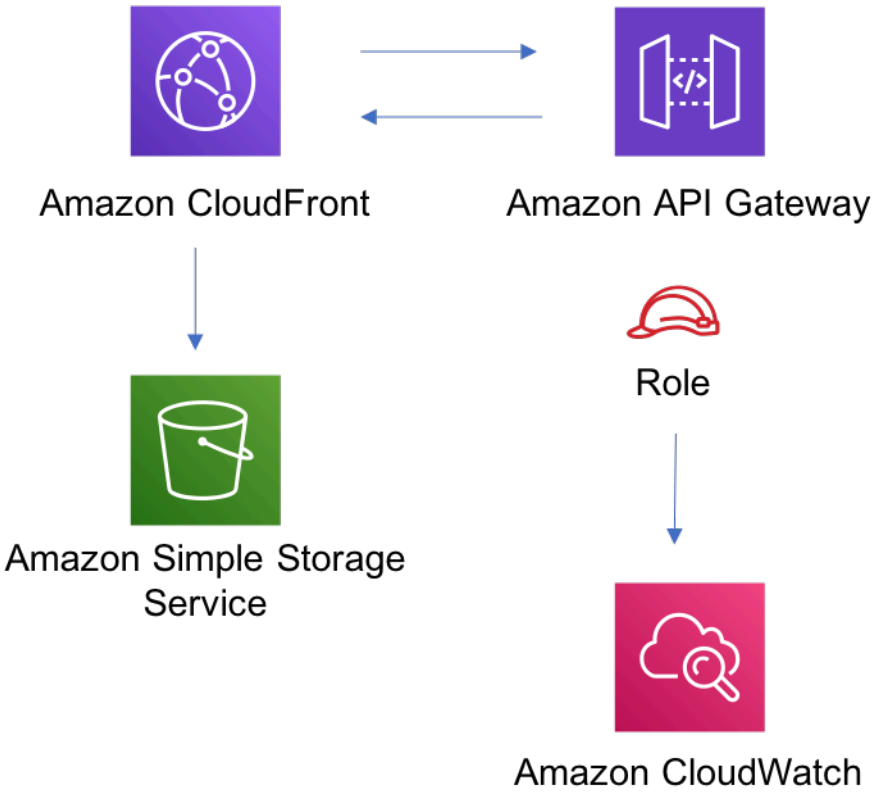
Amazon CloudFront

- Konfigurieren der Zugriffsprotokollierung für CloudFront WebDistribution
- Aktivieren Sie die automatische Einfügung von Best Practice-HTTP-Sicherheitsheadern in allen Antworten von CloudFront WebDistribution

Amazon API Gateway

- Vom Benutzer bereitgestelltes API Gateway -Objekt wird unverändert verwendet
- X-Ray blaufverfolgung aktivieren

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-constructs/aws-cloudfront-apiggateway

aws-cloudfront-apigateway-lambda

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht denSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_cloudfront_apigateway_lambda`
Typoskript	`@aws-solutions-constructs/aws-cloudfront-apigateway-lambda`
Java	`software.amazon.awsconstructs.services.cloudfrontapigatewaylambda`

Overview

Dieses AWS Solutions Construct implementiert eine Amazon CloudFront Distribution vor einer LAMBDA-gesicherten REST-API mit Amazon API Gateway.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`CloudFrontToApiGatewayToLambdaProps`

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts,

Name	Typ	Beschreibung
		die sowohl diese als auch `lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein `existingLambdaObj` wird bereitgestellt.
ApigatewayProps?	`api.LambdaRestApiProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für API Gateway
CloudFrontDistributionProps?	`cloudfront.DistributionProps`	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für die CloudFront Distribution zu überschreiben.
InsertHTTPSecurityHeader?	`boolean`	Optional vom Benutzer bereitgestellte Requisiten zum Ein- und Ausschalten der automatischen Einfügung von Best Practice-HTTP-Sicherheitsheadern in allen Antworten von CloudFront
LogGroupProps?	`logs.LogGroupProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardprops für die CloudWatch Logs-Protokollgruppe.

Muster-Eigenschaften

Name	Typ	Beschreibung
ApigGateway	api.RestApi	Gibt eine Instanz der API Gateway REST-API zurück, die durch das Muster erstellt wurde.
ApigatewayCloudWatchRole	iam.Role	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster erstellt wurde, das die Zugriffsprotokollierung von der API Gateway REST-API zu CloudWatch ermöglicht.
ApigatewayLogGroup	logs.LogGroup	Gibt eine Instanz der Protokoll gruppe zurück, die durch das Muster erstellt wurde, an das API Gateway REST-API-Zugriffsprotokolle gesendet werden.
CloudFrontLoggingBucket?	s3.Bucket	Gibt eine Instanz des Protokoll ierungsbuckets zurück, der durch das Muster für die CloudFront Webverteilung erstellt wurde.
CloudFrontWebDistribution	cloudfront.CloudFrontWebDistribution	Gibt eine Instanz der CloudFront Webverteilung zurück, die durch das Muster erstellt wurde.
EdgeLambdaFunction Version?	lambda.Version	Gibt eine Instanz der Lambda Kantenfunktionsversion

Name	Typ	Beschreibung
		zurück, die durch das Muster erstellt wurde.
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon CloudFront

- Konfigurieren der Zugriffsprotokollierung für CloudFront WebDistribution
- Aktivieren Sie die automatische Einfügung von Best Practice-HTTP-Sicherheitsheadern in allen Antworten von CloudFront WebDistribution

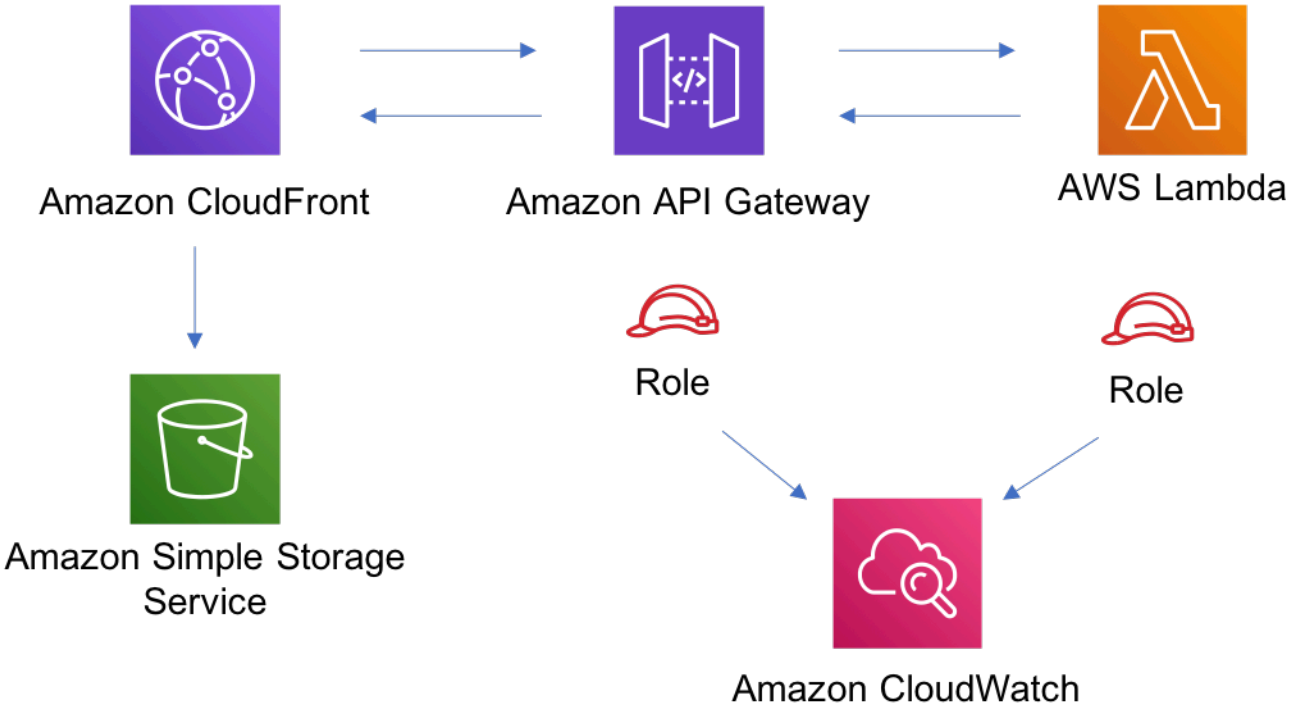
Amazon API Gateway

- Bereitstellen eines regionalen API-Endpunkts
- Aktivieren von CloudWatch Protokollierung für API Gateway
- Konfigurieren der IAM-Rolle für den Zugriff auf die geringste Berechtigung für API Gateway
- Legen Sie den standardmäßigen AuthorizationType für alle API-Methoden auf IAM fest
- X-Ray blaufverfolgung aktivieren

AWS Lambda-Funktion

- IAM-Rolle für den eingeschränkten Privilegitätszugriff für die Lambda -Funktion konfigurieren
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda
- X-Ray blaufverfolgung aktivieren
- Festlegen von Umgebungsvariablen:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-constructs/aws-cloudfront-apigateway-lambda

aws-cloudfront-mediastore

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantic version (Semantic version)Modell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_cloudfront_mediastore`
TypeScript	`@aws-solutions-constructs/aws-cloudfront-mediastore`
Java	`software.amazon.awsconstructs.services.cloudfrontmediastore`

Overview

Dieses AWS Solutions Construct implementiert eine Amazon CloudFront Distribution, die mit einem AWS Elemental MediaStore Container verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct

- `id`string
- props`CloudFrontToMediaStoreProps`

Muster Konstrukt Requisiten

Name	Typ	Beschreibung
ExistingMediaStoreContainerObj?	`mediastore.CfnContainer`	Optionaler, vom Benutzer bereitgestellMediaStore Container zum Überschreiben des standardmäßigen MediaStore-Containers
MediaStoreContainerProps?	`mediastore.CfnContainerProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für den MediaStore Container.
CloudFrontDistributionProps?	`cloudfront.DistributionProps` | `any`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für die CloudFront Distribution.
InsertHTTPSecurityHeader?	`boolean`	Optionale, vom Benutzer bereitgestellte Requisiten zum Ein- und Ausschalten der automatischen Einfügung von Best Practice HTTP-Sicherheitsheadern in allen Antworten von CloudFront.

Muster-Eigenschaften

Name	Typ	Beschreibung
CloudFrontWebDistribution	cloudfront.CloudFrontWebDistribution	Gibt eine Instanz der CloudFront Webverteilung zurück, die durch das Muster erstellt wurde.
MediaStoreContainer	mediastore.CfnContainer	Gibt eine Instanz des MediaStore Containers zurück, der durch das Muster erstellt wurde.
CloudFrontLoggingBucket	s3.Bucket	Gibt eine Instanz des Protokollierungsbuckets zurück, der durch das Muster für die CloudFront Webverteilung erstellt wurde.
CloudFrontOriGinRequestPolicy	cloudfront.OriginRequestPolicy	Gibt eine Instanz der CloudFront Ursprungs anforderungsrichtlinie zurück, die durch das Muster für die CloudFront-Webverteilung erstellt wurde.
CloudFrontOriginAccessIdentity?	cloudfront.OriginAccessIdentity	Gibt eine Instanz der CloudFront Ursprungs-Zugriffsidentität zurück, die durch das Muster für die CloudFront-Webverteilung erstellt wurde.
EdgeLambdaFunctionVersion	lambda.Version	Gibt eine Instanz der Lambda Kantenfunktionsversion zurück, die durch das Muster erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

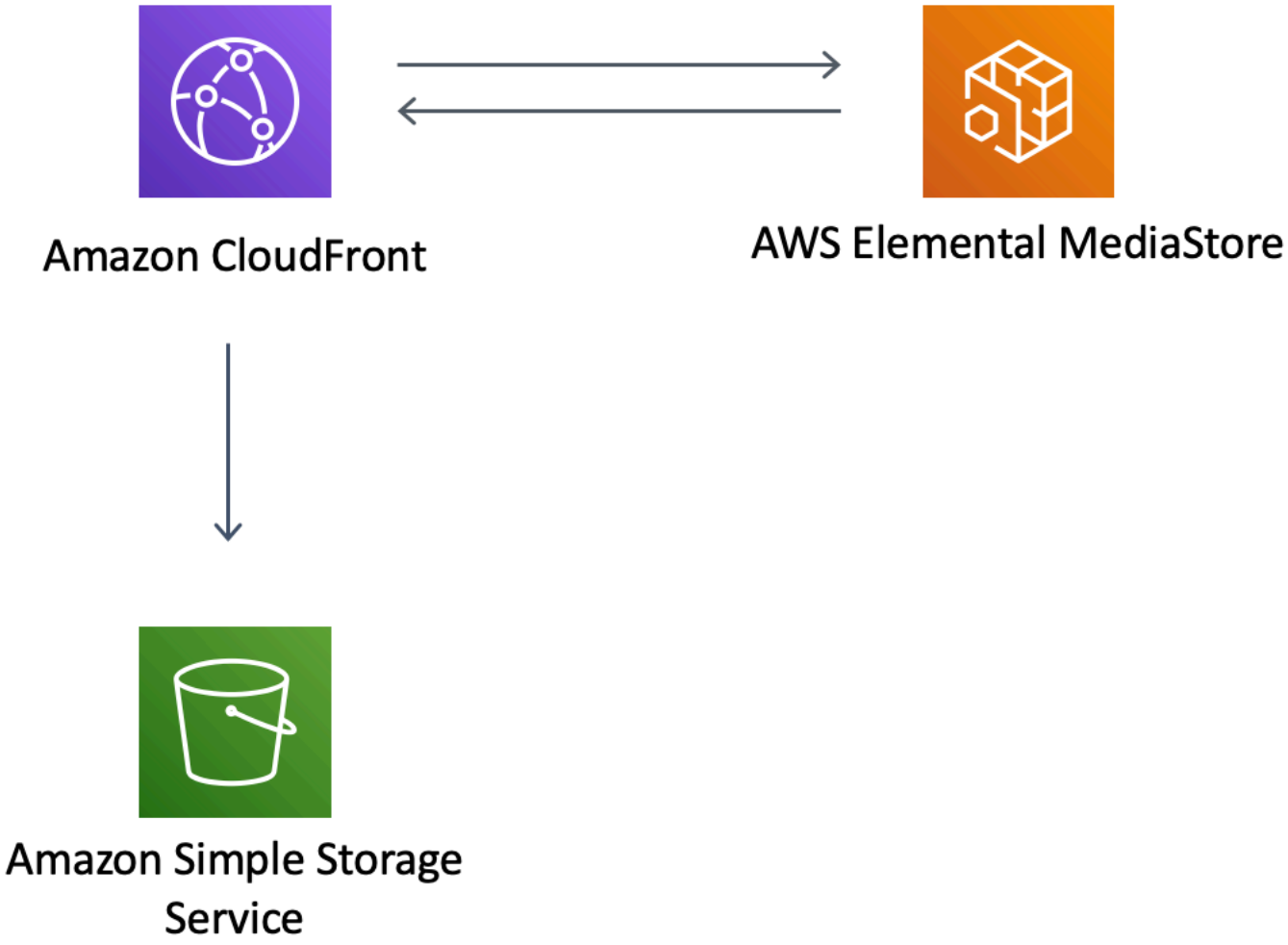
Amazon CloudFront

- Konfigurieren der Zugriffsprotokollierung für die CloudFront Webverteilung
- Aktivieren der CloudFront Ursprungsanforderungsrichtlinie für AWS Elemental MediaStore Container
- Legen Sie fest.`User-Agent`Benutzerdefinierter Header mit CloudFront Ursprungszugriffsi
- Aktivieren Sie die automatische Injektion von Best Practice-HTTP-Sicherheitsheadern in allen Antworten aus der CloudFront Webverteilung

AWS Elemental MediaStore

- Festlegen der Löschrichtlinie, um die Ressource beizubehalten
- Festlegen des Containernamens mit dem CloudFormation -Stacknamen
- Richten Sie den Standard einCross-Origin Resource Sharing (CORS) -Richtlinie
- Richten Sie den Standard einObjektlebenszyklus-Richtlinie
- Richten Sie den Standard einContainerrichtlinie, um nur zu erlaubenaws:UserAgentmit CloudFront Ursprungszugriffsidentität
- Richten Sie den Standard einMetrikrichtlinie
- Aktivieren der Zugriffsprotokollierung

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-cloudfront-mediastore

aws-cloudfront-s3

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_cloudfront_s3`
TypeScript	`@aws-solutions-constructs/aws-cloudfront-s3`
Java	`software.amazon.awsconstructs.services.cloudfronts3`

Overview

Dieses AWS Solutions Construkt stellt eine Amazon CloudFront Verteilung vor einem Amazon S3 Bucket bereit.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`CloudFrontToS3Props`

Muster-Konstrukt

Name	Typ	Beschreibung
Bestehend BuckeObj?	`s3.Bucket`	Vorhandene Instanz des S3 Bucket-Objekts. Wenn dies zur Verfügung gestellt wird, dann wird auch`bucketPro ps` ist ein Fehler.
BucketProps?	`s3.BucketProps`	Optionale vom Benutzer bereitgestellte Eigenscha ften, um die Standarde igenschaften für den Bucket zu überschreiben. Wird ignoriert, wenn ein`existingB ucketObj` wird bereitgestellt.
CloudFrontDistributionProps?	`cloudfront.Distrib utionProps`	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für die CloudFront Distribution zu überschreiben.
InsertHTTPSecurityHeader?	`boolean`	Optional vom Benutzer bereitgestellte Requisiten zum Ein- und Ausschalten der automatischen Einfügung von Best Practice-HTTP-

Name	Typ	Beschreibung
		Sicherheitsheadern in allen Antworten von CloudFront

Muster-Eigenschaften

Name	Typ	Beschreibung
CloudFrontWebDistribution	`cloudfront.CloudFrontWebDistribution`	Gibt eine Instanz der CloudFront Webverteilung zurück, die durch das Muster erstellt wurde.
S3Bucket?	`s3.Bucket`	Gibt eine Instanz des S3-Buckets zurück, der durch das Muster erstellt wurde.
s3loggingBucket?	`s3.Bucket`	Gibt eine Instanz des Protokollierungsbuckets zurück, der durch das Muster für den S3-Bucket erstellt wurde.
EdgeLambdaFunctionVersion?	`lambda.Version`	Gibt eine Instanz der Lambda Kantenfunktionsversion zurück, die durch das Muster erstellt wurde.
CloudFrontLoggingBucket?	`s3.Bucket`	Gibt eine Instanz des Protokollierungsbuckets zurück, der durch das Muster für die CloudFront Webverteilung erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

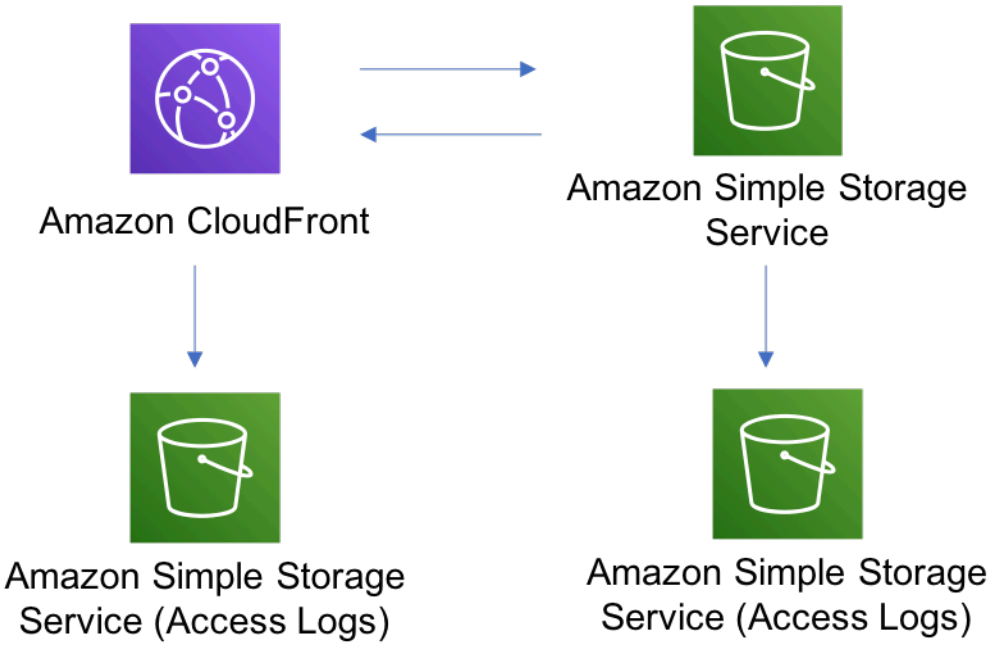
Amazon CloudFront

- Konfigurieren der Zugriffsprotokollierung für CloudFront WebDistribution
- Aktivieren Sie die automatische Einfügung von Best Practice-HTTP-Sicherheitsheadern in allen Antworten von CloudFront WebDistribution

Amazon S3 Bucket

- Konfigurieren der Zugriffsprotokollierung für S3 Bucket
- Aktivieren der serverseitigen Verschlüsselung für S3 Bucket mit AWS verwaltetem KMS-Schlüssel
- Aktivieren der Versionierung für S3 Bucket
- Öffentlichen Zugriff für S3 Bucket nicht zulassen
- Beibehalten des S3-Bucket beim Löschen des CloudFormation -Stacks
- Erzwingen der Verschlüsselung von Daten während der Übertragung
- Wendet Lebenszyklusregel an, um nicht aktuelle Objektversionen nach 90 Tagen in Glacier-Speicher zu verschieben

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-constructs/aws-cloudfront-s3

aws-cognito-apigateway-lambda

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht denSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_cognito_apigateway_lambda`
Typoskript	`@aws-solutions-constructs/aws-cognito-apigateway-lambda`
Java	`software.amazon.awsconstructs.services.cognitoapigatewaylambda`

Overview

Dieses AWS Solutions Construct implementiert Amazon Cognito zum Sichern einer LAMBDA-gesicherten REST-API mit Amazon API Gateway.

Hier ist eine minimale bereitstellbare Musterdefinition 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'
    }
});
```

Wenn Sie Ressourcen und Methoden in Ihrer API definieren (z.`proxy = false`) müssen Sie die`addAuthorizers()`-Methode, nachdem die API vollständig definiert ist. Dadurch wird sichergestellt, dass jede Methode in Ihrer API geschützt ist.

Hier ist ein Beispiel 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`CognitoToApiGatewayToLambdaProps`

Muster-Konstrukt

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch `lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein `existingLambdaObj` wird bereitgestellt.
ApigatewayProps?	`api.LambdaRestApiProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für API Gateway
CognitouserPoolProps?	`cognito.UserPoolProps`	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für den Cognito Benutzerpool
CognitouserPoolClientProps?	`cognito.UserPoolClientProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für den Cognito User Pool Client
LogGroupProps?	`logs.LogGroupProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardp

Name	Typ	Beschreibung
		rops für die CloudWatch Logs s-Protokollgruppe.

Muster-Eigenschaften

Name	Typ	Beschreibung
ApiGateway	`api.RestApi`	Gibt eine Instanz der API Gateway REST-API zurück, die durch das Muster erstellt wurde.
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
userPool	`cognito.UserPool`	Gibt eine Instanz des Cognito Benutzerpools zurück, die durch das Muster erstellt wurde.
UserPoolClient	`cognito.UserPoolClient`	Gibt eine Instanz des Cognito Benutzerpool-Clients zurück, die durch das Muster erstellt wurde.
ApigatewayCloudWatchRole	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster erstellt wurde, das die Zugriffsprotokollierung von der API Gateway REST-API zu CloudWatch ermöglicht.
ApigatewayLogGroup	`logs.LogGroup`	Gibt eine Instanz der Protokoll gruppe zurück, die durch das

Name	Typ	Beschreibung
		Muster erstellt wurde, an das API Gateway REST-API-Zugriffsprotokolle gesendet werden.
ApigatewayAuthorizer	`api.CfnAuthorizer`	Gibt eine Instanz des API Gateway Autorizers zurück, der durch das Muster erstellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon Cognito

- Festlegen einer Kennwortrichtlinie für -
- Erzwingen des erweiterten Sicherheitsmodus für Benutzerpools

Amazon API Gateway

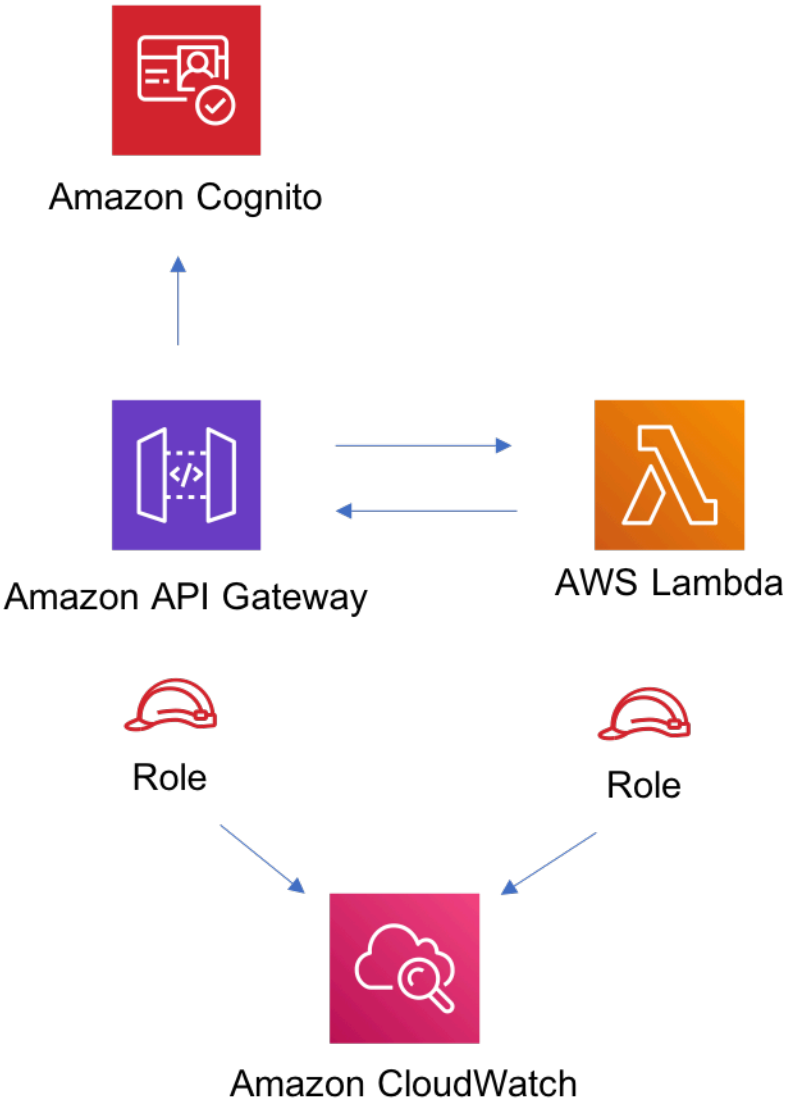
- Bereitstellen eines Edge-optimierten API-Endpunkts
- Aktivieren der CloudWatch -Protokollierung für API Gateway
- Konfigurieren der IAM-Rolle für den Zugriff auf die geringste Berechtigung für API Gateway
- Legen Sie den standardmäßigen AuthorizationType für alle API-Methoden auf IAM fest
- Aktivieren der X-Ray blaufverfolgung

AWS Lambda-Funktion

- Konfigurieren der IAM-Rolle für den eingeschränkten Berechtigungszugriff für die Lambda -Funktion
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda
- Aktivieren der X-Ray blaufverfolgung

- Festlegen von Umgebungsvariablen:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-konstrukte/aws-cognito-apigateway-lambda

AWS-Dynamodb-Stream-Lambda

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht denSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_dynamodb_stream_lambda`
Typoskript	`@aws-solutions-constructs/aws-dynamodb-stream-lambda`
Java	`software.amazon.awsconstructs.services.dynamodbstreamlambda`

Overview

Dieses AWS Solutions Construct implementiert eine Amazon DynamoDB -Tabelle mit Stream, um die AWS Lambda Funktion mit den am wenigsten privilegierten Berechtigungen aufzurufen.

Hier ist eine minimale bereitstellbare Musterdefinition:

```
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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`DynamoDBStreamToLambdaProps`

Muster-Konstrukt

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird bereitgestellt.

Name	Typ	Beschreibung
DynamoTableProps?	dynamodb.TableProps	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für DynamoDB -Tabelle
ExistingTableObj?	dynamodb.Table	Vorhandene Instanz des DynamoDB -Tabellen objekts, die sowohl diese als auchdynamoTableProps führt zu einem -Fehler.
DynamoEventSourceProps?	aws-lambda-event-sources.DynamoEventSourceProps	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für die DynamoDB Ereignisquelle

Muster-Eigenschaften

Name	Typ	Beschreibung
DynamoTable	dynamodb.Table	Gibt eine Instanz der DynamoDB -Tabelle zurück, die durch das Muster erstellt wurde.
LambdaFunction	lambda.Function	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.

Lambda-Funktion

Dieses Muster erfordert eine Lambda Funktion, die Daten aus dem DynamoDB Stream in den Elasticsearch-Dienst posten kann. Eine Beispielfunktion wird bereitgestelltHier gilt.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

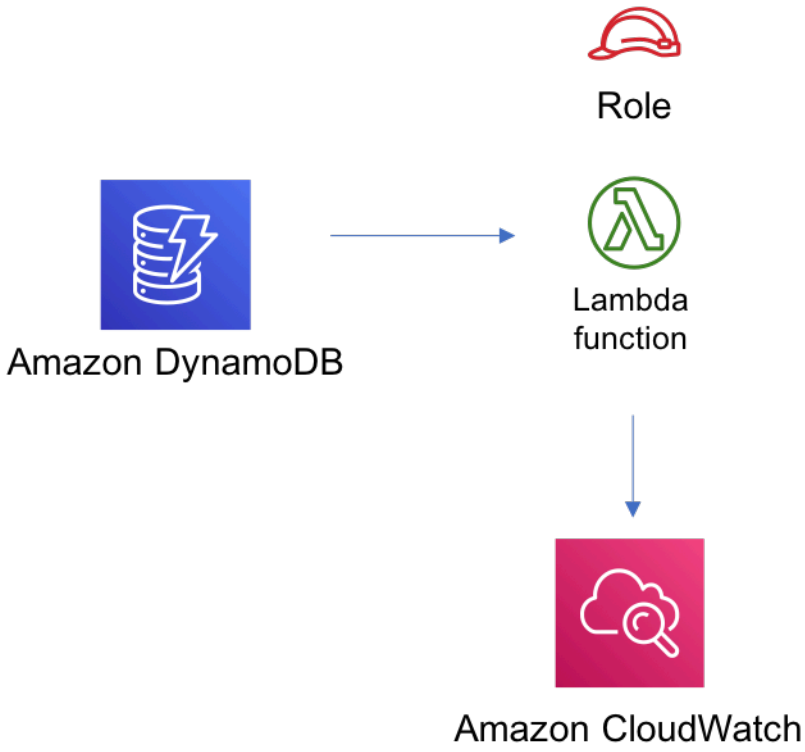
Amazon DynamoDB Tabelle

- Festlegen des Abrechnungsmodus für DynamoDB -Tabelle auf On-Demand (Bezahlung pro Anforderung)
- Aktivieren der serverseitigen Verschlüsselung für DynamoDB Table mit AWS verwaltetem KMS-Schlüssel
- Erstellt einen Partitionsschlüssel namens 'id' für DynamoDB -Tabelle
- Beibehalten der Tabelle beim Löschen des CloudFormation -Stacks
- Kontinuierliche Sicherungen und zeitpunktbezogene Wiederherstellung

AWS Lambda-Funktion

- IAM-Rolle für den eingeschränkten Privilegitätszugriff für die Lambda Funktion konfigurieren
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda
- X-Ray blaufverfolgung aktivieren
- Funktionen zur Fehlerbehandlung aktivieren: Bisect bei Funktion aktivieren Fehler; Standardeinstellung Maximales Datensatzalter (24 Stunden); Standardeinstellung Maximale Wiederholungsversuche (500) festlegen und SQS-Warteschlange für Dead-Letter als Ziel bei einem Fehler bereitstellen
- Festlegen von Umgebungsvariablen:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-konstrukte/aws-dynamodb-stream-lambda

aws-dynamodb-stream-lambda-elasticsearch-kibana

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantic VersioningModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_dynamodb_stream_lambda_elasticsearch_kibana`
Typographische Schrift	`@aws-solutions-constructs/aws-dynamodb-stream-lambda-elasticsearch-kibana`
Java	`software.amazon.awsconstructs.services.dynamodbstreamlambdaelasticsearchkibana`

Overview

Dieses AWS Solutions Construct implementiert Amazon DynamoDB -Tabelle mit Stream, einer AWS Lambda Funktion und einem Amazon Elasticsearch Service mit den geringsten Berechtigungen.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`DynamoDBStreamToLambdaToElasticSearchAndKibanaProps`

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird zur Verfügung gestellt.

Name	Typ	Beschreibung
DynamoTableProps?	dynamodb.TableProps	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für DynamoDB -Tabelle
ExistingTableObj?	dynamodb.Table	Vorhandene Instanz des DynamoDB -Tabellen objekts, die sowohl diese als auchdynamoTableProps führt zu einem -Fehler.
DynamoEventSourceProps?	aws-lambda-event-sources.DynamoEventSourceProps	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für die DynamoDB Ereignisquelle
ESDomainProps?	elasticsearch.CfnDomainProps	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für den Amazon Elasticsearch Service
domainName	string	Domänenname für den Cognito und den Amazon Elasticsearch Service
CreateCloudWatchalarms	boolean	Gibt an, ob empfohlene CloudWatch Alarme erstellt werden sollen.

Muster-Eigenschaften

Name	Typ	Beschreibung
CloudwatChalarms?	cloudwatch.Alarm[]	Gibt eine Liste von einem oder mehreren CloudWatch Alarmen zurück, die durch das Muster erstellt wurden.
DynamoTable	dynamodb.Table	Gibt eine Instanz der DynamoDB -Tabelle zurück, die durch das Muster erstellt wurde.
ElasticSearchDomain	elasticsearch.CfnDomain	Gibt eine Instanz der Elasticsearch-Domäne zurück, die durch das Muster erstellt wurde.
IdentityPool	cognito.CfnIdentityPool	Gibt eine Instanz des Cognito Identitätspools zurück, der durch das Muster erstellt wurde.
LambdaFunction	lambda.Function	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
userPool	cognito.UserPool	Gibt eine Instanz des Cognito Benutzerpools zurück, die durch das Muster erstellt wurde.
UserPoolClient	cognito.UserPoolClient	Gibt eine Instanz des Cognito Benutzerpool-Clients zurück, die durch das Muster erstellt wurde.

Lambda-Funktion

Dieses Muster erfordert eine Lambda Funktion, die Daten aus dem DynamoDB Stream in den Elasticsearch-Dienst posten kann. Eine Beispielfunktion wird bereitgestelltHier gilt.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon DynamoDB Tabelle

- Festlegen des Abrechnungsmodus für DynamoDB -Tabelle auf On-Demand (Bezahlung pro Anforderung)
- Aktivieren der serverseitigen Verschlüsselung für DynamoDB Table mit AWS verwaltetem KMS-Schlüssel
- Erstellt einen Partitionsschlüssel namens 'id' für DynamoDB -Tabelle
- Beibehalten der Tabelle beim Löschen des CloudFormation -Stacks
- Ermöglicht kontinuierliche Sicherungen und zeitpunktbezogene Wiederherstellung

AWS Lambda-Funktion

- IAM-Rolle für den eingeschränkten Zugriff auf Berechtigungen für die Lambda -Funktion konfigurieren
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion
- Aktivieren der X-Ray blaufverfolgung
- Funktionen zur Fehlerbehandlung aktivieren: Bisect bei Funktion aktivieren Fehler; Standardeinstellung Maximales Datensatzalter (24 Stunden); Standardeinstellung Maximale Wiederholungsversuche (500) festlegen und SQS-Warteschlange für Dead-Letter als Ziel bei einem Fehler bereitstellen
- Festlegen von Umgebungsvariablen:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Amazon Cognito

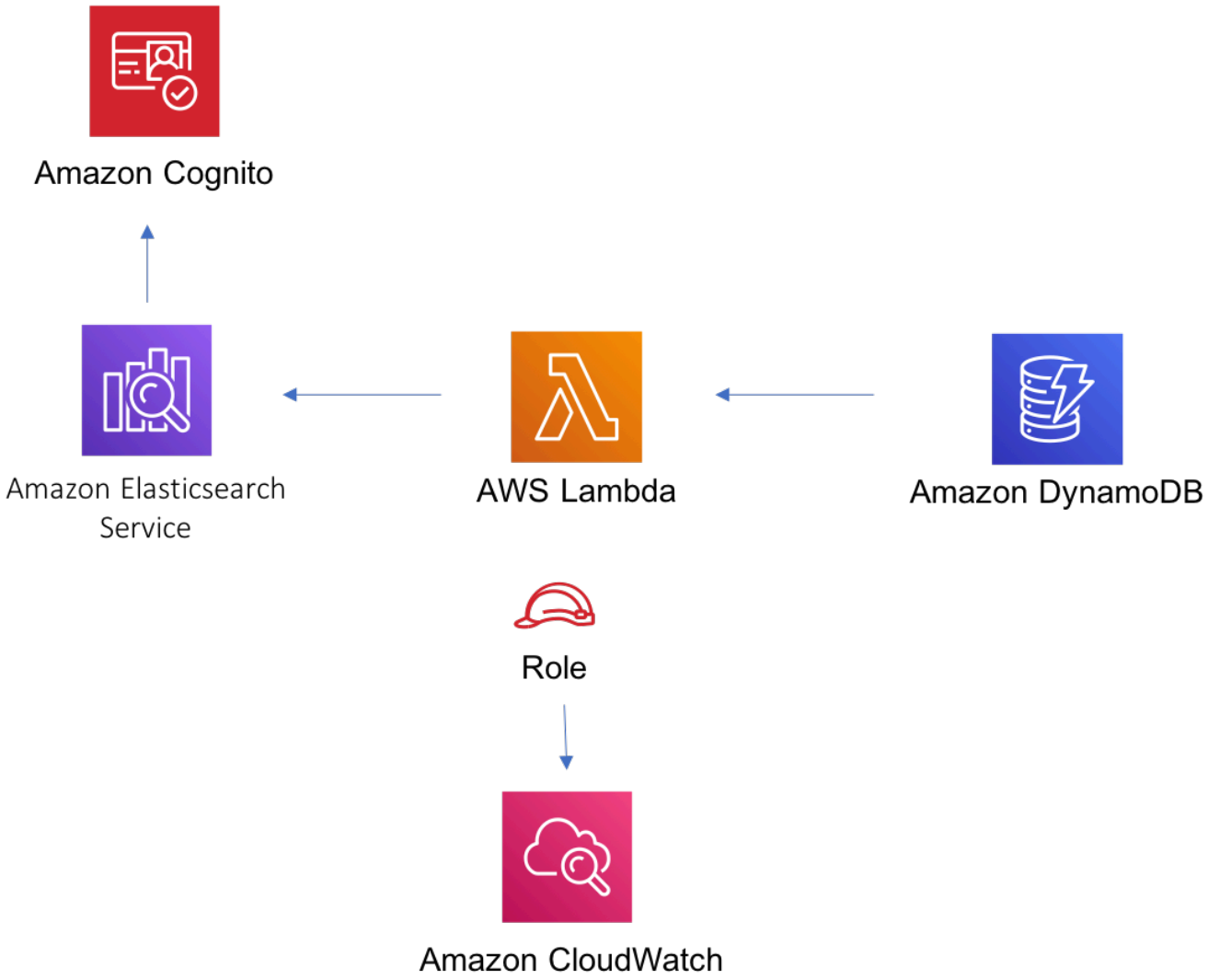
- Festlegen von Kennwortrichtlinie für -

- Erzwingen des erweiterten Sicherheitsmodus für Benutzerpools

Amazon Elasticsearch Service

- Bereitstellen von Best Practices CloudWatch Alarmen für die Elasticsearch-Domäne
- Sichern des Kibana-Dashboard-Zugriffs mit Cognito User
- Aktivieren der serverseitigen Verschlüsselung für Elasticsearch Domain mit AWS verwaltetem KMS-Schlüssel
- Aktivieren Sie die Knoten-zu-Knoten-Verschlüsselung für Elasticsearch-Domäne
- Konfigurieren des Clusters für die Amazon ES-Domäne

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:	
	@aws -solutions-konstrukte/aws-dynamodb-stream-lambda-elasticsearch-kibana

aws-events-regel-kinesisfireschlauch-s3

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantic version (Semantic version)Modell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_events_rule_kinesisfirehose_s3`
Typecript	`@aws-solutions-constructs/aws-events-rule-kinesisfirehose-s3`
Java	`software.amazon.awsconstructs.services.eventsrulekinesisfirehoses3`

Overview

Dieses AWS Solutions Construct implementiert eine Amazon CloudWatch Events regel, um Daten an einen Amazon Kinesis Data Firehose Bereitstellungsstream zu senden, der mit einem Amazon S3 Bucket verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`EventsRuleToKinesisFirehoseToS3Props`

Muster-Konstrukt

Name	Typ	Beschreibung
EventRuleProps	`events.RuleProps`	Vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die CloudWatch Ereignisregel zu überschreiben.
KinesisFirehoseProps?	`aws-kinesisfirehose.CfnDeliveryStreamProps`	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für Kinesis Firehose Delivery Stream zu überschreiben.
Bestehend BuckeObj?	`s3.IBucket`	Vorhandene Instanz des S3 Bucket-Objekts Wenn dies zur Verfügung gestellt wird, dann wird auch`bucketProps` ist ein Fehler.
BucketProps?	`s3.BucketProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für den S3-Bucket.
LogGroupProps?	`logs.LogGroupProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardp rops für die CloudWatch Logs s-Protokollgruppe

Muster-Eigenschaften

Name	Typ	Beschreibung
EreignissRegel	events.Rule	Gibt eine Instanz der Ereignisregel zurück, die durch das Muster erstellt wurde.
KinesisFirehose	kinesisfirehose.CfnDeliveryStream	Gibt eine Instanz des Kinesis Firehose Übermittlungsdatenstroms zurück, der durch das Muster erstellt wurde.
S3Bucket	s3.Bucket	Gibt eine Instanz des S3-Buckets zurück, der durch das Muster erstellt wurde.
s3loggingBucket?	s3.Bucket	Gibt eine Instanz des Protokollierungsbuckets zurück, der durch das Muster für den S3-Bucket erstellt wurde.
EventsRolle?	iam.Role	Gibt eine Instanz der Rolle zurück, die vom Konstrukt für die CloudWatch Ereignisregel erstellt wurde.
KinesisFirehoserole	iam.Role	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster für den Kinesis Firehose-Bereitstellungsstream erstellt wurde.
KinesisFirehoselogGruppe	logs.LogGroup	Gibt eine Instanz der Protokollgruppe zurück, die durch das Muster erstellt wurde, an das Kinesis Firehose-Zugriffsprotokolle gesendet werden.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Regel für Amazon CloudWatch Events

- Konfigurieren Sie die IAM-Rolle für den Zugriff auf die geringste Berechtigung für Ereignisregel, um im Kinesis Firehose Delivery Stream zu veröffentlichen.

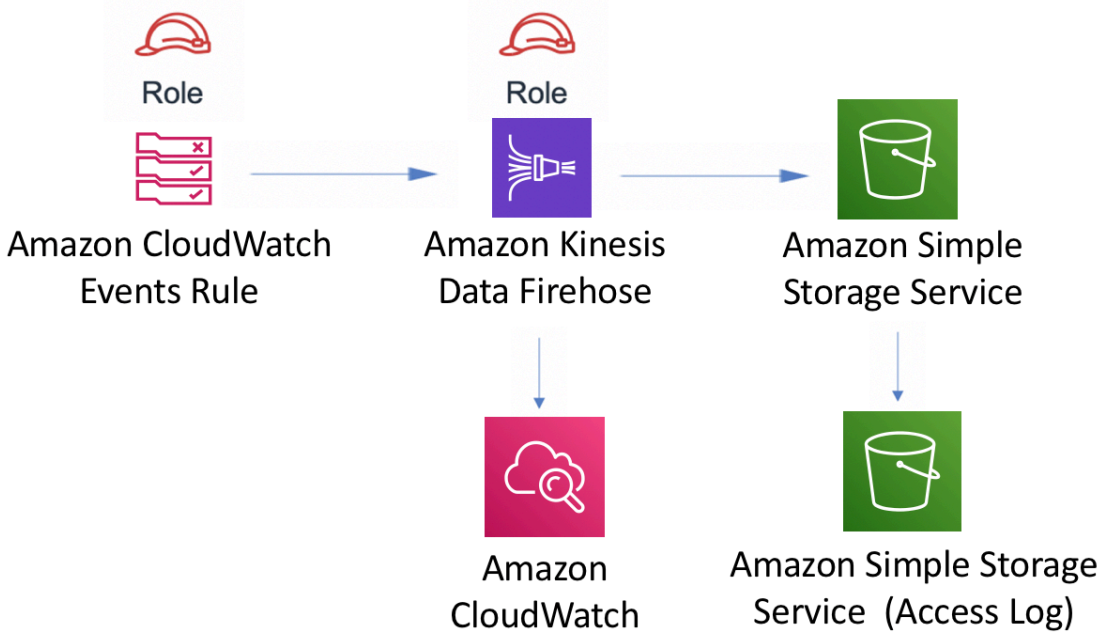
Amazon Kinesis Firehose

- Aktivieren Sie die CloudWatch Protokollierung für Kinesis Firehose.
- Konfigurieren der IAM-Rolle für den Zugriff auf die geringste Berechtigung für Amazon Kinesis Firehose

Amazon S3 Bucket

- Konfigurieren Sie die Zugriffsprotokollierung für Bucket.
- Aktivieren Sie die serverseitige Verschlüsselung für Bucket mithilfe von AWS verwaltetem KMS-Schlüssel.
- Aktivieren der Versionierung für den Bucket
- Erlauben Sie keinen öffentlichen Zugriff für den Bucket.
- Behalten Sie den Bucket beim Löschen des CloudFormation-Stacks bei.
- Wendet die Lebenszyklusregel an, um nicht aktuelle Objektversionen nach 90 Tagen in den Glacier-Speicher zu verschieben.

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-konstrukte/aws-events-rule-kinesisfireschlauch-s3

aws-events-regel-kinesisstreams

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_events_rule_kinesisstream`
Typoskript	`@aws-solutions-constructs/aws-events-rule-kinesisstreams`
Java	`software.amazon.awsconstructs.services.eventsrulekinesisstream`

Overview

Dieses AWS Solutions Construct implementiert eine Amazon CloudWatch Events regel zum Senden von Daten an einen Amazon Kinesis Datenstrom.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsEventsRuleToKinesisStreamsProps

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
EventRuleProps	events.RuleProps	Vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die CloudWatch Ereignisregel zu überschreiben.
Bestehend StreamObj?	kinesis.Stream	Vorhandene Instanz von Kinesis Stream, die sowohl diese als auchkinesisStreamProps führt zu einem -Fehler.
KinesisStreamProps?	kinesis.StreamProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für den Kinesis Stream.

Name	Typ	Beschreibung
CreateCloudWatchalarms	`boolean`	Gibt an, ob empfohlene CloudWatch Alarme erstellt werden sollen.

Muster-Eigenschaften

Name	Typ	Beschreibung
EreignissRegel	`events.Rule`	Gibt eine Instanz der Ereignisr egel zurück, die durch das Muster erstellt wurde.
KinesisStream-Version	`kinesis.Stream`	Gibt eine Instanz des Kinesis Streams zurück, der durch das Muster erstellt wurde.
EventsRolle?	`iam.Role`	Gibt eine Instanz der Rolle zurück, die vom Konstrukt für die CloudWatch Ereignisregel erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Regel für Amazon CloudWatch Events

- Konfigurieren Sie die IAM-Rolle für den Zugriff auf die geringste Berechtigung für Ereignisregel, die im Kinesis Datenstrom veröffentlicht werden soll.

Amazon Kinesis Streams

- Aktivieren Sie die serverseitige Verschlüsselung für Kinesis Data Stream mit AWS Managed KMS Key.

Architecture

GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-events-rule-kinesisstreams

aws-events-regel-Lambda

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantic VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_events_rule_lambda`
Typoskript	`@aws-solutions-constructs/aws-events-rule-lambda`
Java	`software.amazon.awsconstructs.services.eventsrulelambda`

Overview

Diese AWS Solutions Construct implementiert eine AWS-Ereignisregel und eine AWS Lambda-Funktion.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`EventsRuleToLambdaProps`

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird bereitgestellt.

Name	Typ	Beschreibung
EventRuleProps	events.RuleProps	Vom Benutzer bereitgestellte EventRuleProps, um die Standardwerte zu überschre iben

Muster-Eigenschaften

Name	Typ	Beschreibung
EreignissRegel	events.Rule	Gibt eine Instanz der Ereignisr egel zurück, die durch das Muster erstellt wurde.
LambdaFunction	lambda.Function	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon CloudWatch Events regel

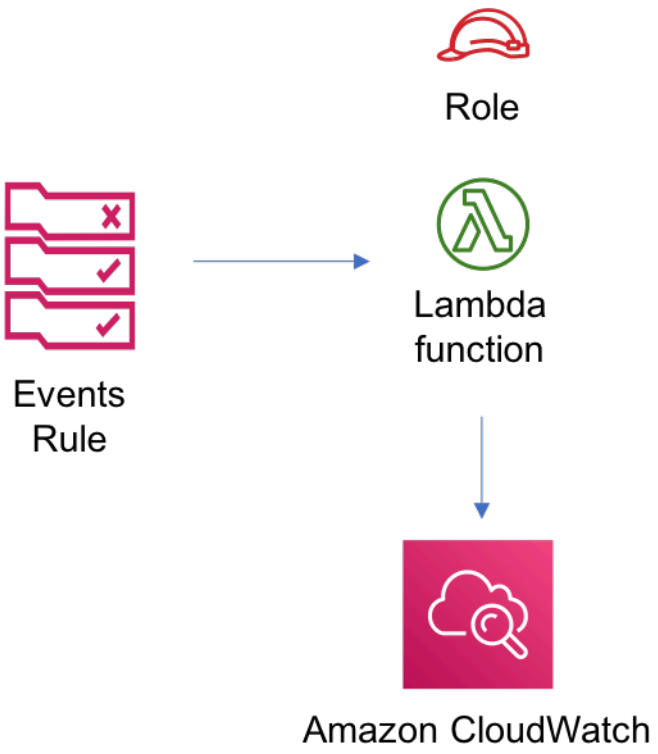
- Gewähren Sie CloudWatch Ereignissen geringste Berechtigungen, um die Lambda Funktion auszulösen

AWS Lambda-Funktion

- IAM-Rolle für den eingeschränkten Zugriff auf Berechtigungen für die Lambda -Funktion
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion
- Aktivieren von X-Ray blaufverfolgung
- Festlegen von Umgebungsvariablen:

- `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-konstrukte/aws-events-rule-lambda

aws-events-rule-sns

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische

VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_events_rule_sns`
Typoskript	`@aws-solutions-constructs/aws-events-rule-sns`
Java	`software.amazon.awsconstructs.services.eventsrulesns`

Overview

Mit diesem Muster wird eine Amazon CloudWatch Events -Regel implementiert, die mit einem Amazon SNS -Thema verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition:

```
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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`EventsRuleToSnsProps`

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
EventRuleProps	`events.RuleProps`	Vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die CloudWatch Ereignisregel zu überschreiben.
Bestehend TopicObj?	`sns.Topic`	Vorhandene Instanz des SNS Topic-Objekts, die sowohl diese als auch`topicProps` führt zu einem -Fehler.

Name	Typ	Beschreibung
TopicProps?	`sns.TopicProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für das SNS-Thema zu überschreiben. Wird ignoriert, wenn ein `existingTopicObj` wird bereitgestellt.
EnableEncryptionWithCustomerManagedKey?	`boolean`	Gibt an, ob ein vom Kunden verwalteter Verschlüsselungsschlüssel verwendet werden soll, der entweder von dieser CDK-App verwaltet oder importiert wird. Beim Importieren eines Verschlüsselungsschlüssels muss dieser im Feld `encryptionKey` -Eigenschaft für dieses Konstrukt.
encryptionKey?	`kms.Key`	Ein optionaler, vorhandener Verschlüsselungsschlüssel, der anstelle des Standard-Verschlüsselungsschlüssels verwendet werden soll.
EncryptionKeyProps?	`kms.KeyProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für den Verschlüsselungsschlüssel zu überschreiben.

Muster-Eigenschaften

Name	Typ	Beschreibung
EreignissRegel	`events.Rule`	Gibt eine Instanz der Ereignisregel zurück, die durch das Muster erstellt wurde.
snsTopic	`sns.Topic`	Gibt eine Instanz des SNS-Themas zurück, das durch das Muster erstellt wurde.
encryptionKey	`kms.Key`	Gibt eine Instanz des Verschlüsselungsschlüssels zurück, der durch das Muster erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

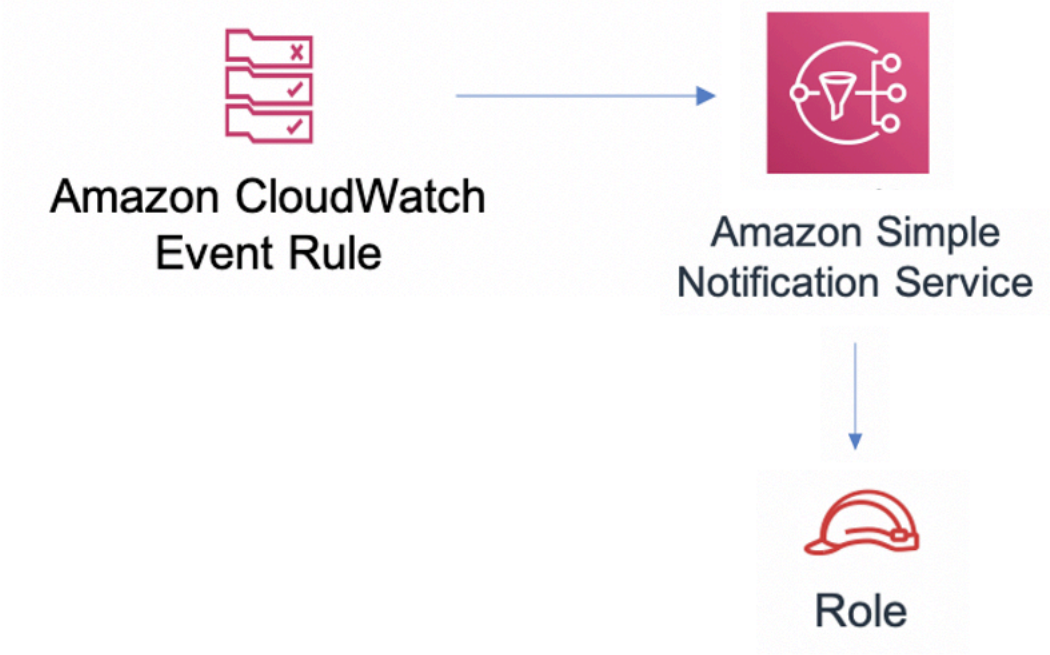
Amazon CloudWatch Events

- Erteilen Sie CloudWatch Ereignissen die geringsten Berechtigungen, um sie im SNS-Thema zu veröffentlichen.

Amazon SNS-Thema

- Konfigurieren Sie Zugriffsberechtigungen mit den geringsten Berechtigungen für das SNS-Thema.
- Aktivieren der serverseitigen Verschlüsselung für SNS-Thema mithilfe des vom Kunden verwalteten AWS KMS-Schlüssels.
- Erzwingen der Verschlüsselung von Daten während der Übertragung

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-konstrukte/aws-events-rule-sns

aws-events-regel-sqs

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht denSemantische

VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_events_rule_sqs`
Typoskript	`@aws-solutions-constructs/aws-events-rule-sqs`
Java	`software.amazon.awsconstructs.services.eventsrulesqs`

Overview

Dieses Muster implementiert eine Amazon CloudWatch Events, die mit einer Amazon SQS Warteschlange verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition:

```
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);
```

Parameter

- BereichConstruct
- idstring
- propsEventsRuleToSqsProps

Muster-Konstrukt

Name	Typ	Beschreibung
EventRuleProps	events.RuleProps	Vom Benutzer bereitges tellte Eigenschaften, um die Standardeigenschaften für die CloudWatch Ereignisregel zu überschreiben.
BestehenQueueObj?	sqs.Queue	Eine optionale, vorhanden e SQS-Warteschlange, die anstelle der Standardw arteschlange verwendet

Name	Typ	Beschreibung
		werden soll. Bereitste llung sowohl dieser als auchqueueProps führt zu einem -Fehler.
QueueProps?	sqs.QueueProps	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die SQS-Warteschlange zu überschreiben. Wird ignoriert, wenn einexistingQueueObj wird bereitgestellt.
EnableQueueSurging?	boolean	Gibt an, ob der Lambda Funktion zusätzliche Berechtig ungen erteilt werden sollen, um die SQS-Warteschlange zu bereinigen. Standarde instellung: false.
DeployDeadletterQueue?	boolean	Erstellen Sie eine sekundäre Warteschlange, die als Warteschlange für unzustell bare Nachrichten verwendet werden soll. Standarde instellung: true.
DeadLetterQueueProps?	sqs.QueueProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardr equisiten für die Warteschl ange für tote Buchstaben. Wird nur verwendet, wenn diedeployDeadLetterQueue auf true festgelegt.

Name	Typ	Beschreibung
MaxReceiveCount?	`number`	Die Anzahl, wie oft eine Nachricht nicht erfolgreich entfernt werden kann, bevor sie in die Warteschlange für unzustellbare Nachrichten verschoben wird. Standarde instellung: `15`.
EnableEncryptionWithCustomerManagedKey?	`boolean`	Gibt an, ob ein vom Kunden verwalteter Verschlüsselungsschlüssel verwendet werden soll, der entweder von dieser CDK-App verwaltet oder importiert wird. Beim Importieren eines Verschlüsselungsschlüssels muss dieser in der`encryptionKey` -Eigenschaft für dieses Konstrukt.
encryptionKey?	`kms.Key`	Ein optionaler, vorhandener Verschlüsselungsschlüssel, der anstelle des Standard-Verschlüsselungsschlüssels verwendet werden soll.
EncryptionKeyProps?	`kms.KeyProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für den Verschlüsselungsschlüssel zu überschreiben.

Muster-Eigenschaften

Name	Typ	Beschreibung
EreignissRegel	`events.Rule`	Gibt eine Instanz der Ereignisregel zurück, die durch das Muster erstellt wurde.
SQSQueue	`sqs.Queue`	Gibt eine Instanz der SQS-Warteschlange zurück, die durch das Muster erstellt wurde.
encryptionKey	`kms.Key`	Gibt eine Instanz des Verschlüsselungsschlüssels zurück, der durch das Muster erstellt wurde.
DeadLetterQueue?	`sqs.Queue`	Gibt eine Instanz der Warteschlange für toten Buchstaben zurück, die durch das Muster erstellt wurde, wenn eine bereitgestellt wird.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon CloudWatch Events

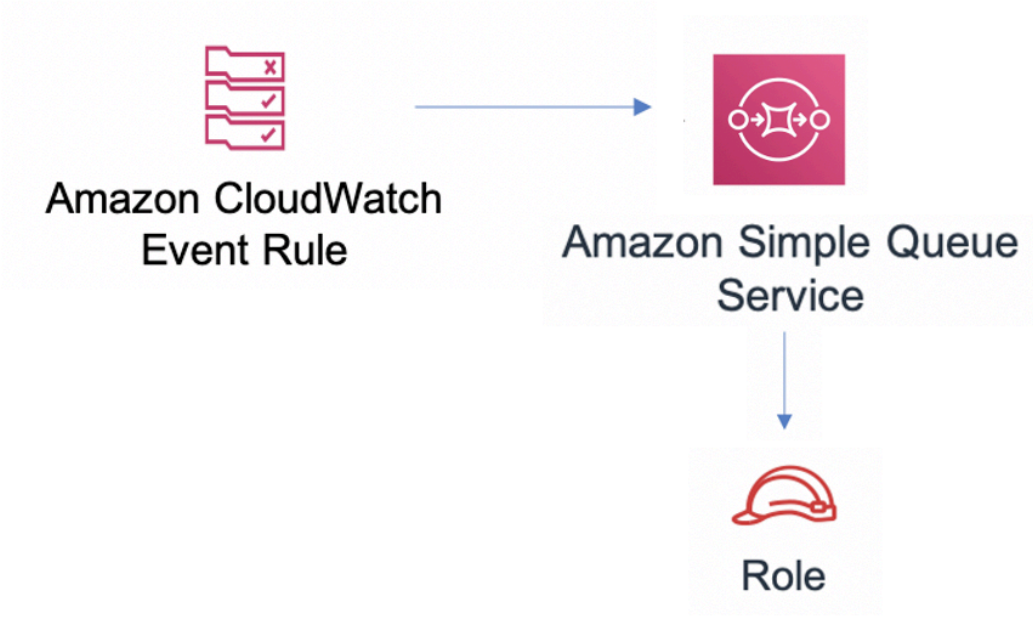
- Erteilen Sie CloudWatch Ereignissen die geringsten Berechtigungen, um sie in der SQS-Warteschlange zu veröffentlichen.

Amazon SQS-Warteschlange

- Bereitstellen einer Warteschlange für unzustellbare Nachrichten für die Quellwarteschlange

- Aktivieren Sie die serverseitige Verschlüsselung für die Quellwarteschlange mithilfe eines vom Kunden verwalteten AWS KMS-Schlüssels.
- Erzwingen der Verschlüsselung von Daten während der Übertragung

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

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

aws-events-regel-Schritt-Funktion

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_events_rule_step_function`
TypeScript	`@aws-solutions-constructs/aws-events-rule-step-function`
Java	`software.amazon.awsconstructs.services.eventsrulestepfunction`

Overview

Dieses AWS Solutions Construct implementiert eine AWS Ereignisregel und eine AWS Step-Funktion.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`EventsRuleToStepFunctionProps`

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
StateMachineProps	`sfn.StateMachineProps`	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für SFN.State Machine
EventRuleProps	`events.RuleProps`	Vom Benutzer bereitgestellte EventRuleProps, um die Standardwerte zu überschreiben

Name	Typ	Beschreibung
CreateCloudWatchalarms	`boolean`	Gibt an, ob empfohlene CloudWatch Alarme erstellt werden sollen.
LogGroupProps?	`logs.LogGroupProps`	Optionale, vom Benutzer bereitgestellte Requisiten, um die Standardprops für die CloudWatch Logs s-Protoko llgruppe zu überschreiben.

Muster-Eigenschaften

Name	Typ	Beschreibung
CloudWatchalarms?	`cloudwatch.Alarm[]`	Gibt eine Liste von einem oder mehreren CloudWatch Alarmen zurück, die durch das Muster erstellt wurden.
EreignissRegel	`events.Rule`	Gibt eine Instanz der Ereignisr egel zurück, die durch das Muster erstellt wurde.
StateMachine	`sfn.StateMachine`	Gibt eine Instanz der Zustandsmaschine zurück, die durch das Muster erstellt wurde.
StateMachineLogGroup	`logs.LogGroup`	Gibt eine Instanz der Protokoll gruppe zurück, die durch das Muster für den Zustandsm aschine erstellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

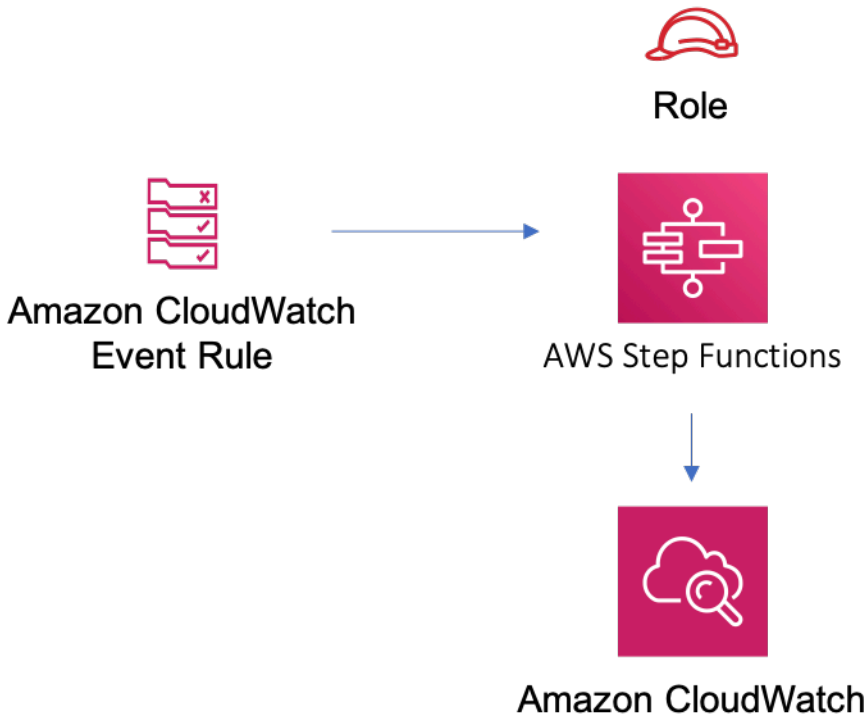
Amazon CloudWatch Events regel

- Gewähren Sie CloudWatch Ereignissen geringste Berechtigungen, um die Lambda Funktion auszulösen

AWS Step Function

- Aktivieren von CloudWatch Protokollierung für API Gateway
- Bereitstellen von bewährten CloudWatch Alarmen für die Step-Funktion

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-events-rule-step-Funktion

aws-iot-kinesisfireschlauch-s3

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_iot_kinesisfirehose_s3`
Typoskript	`@aws-solutions-constructs/aws-iot-kinesisfirehose-s3`
Java	`software.amazon.awsconstructs.services.iotkinesisfirehoses3`

Overview

Dieses AWS Solutions Construct implementiert eine AWS IoT MQTT-Themenregel zum Senden von Daten an einen Amazon Kinesis Data Firehose Bereitstellungsstream, der mit einem Amazon S3 Bucket verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`IotToKinesisFirehoseToS3Props`

Muster-Requisiten

Name	Typ	Beschreibung
IottopicRuleProps	`iot.CfnTopicRuleProps`	Vom Benutzer bereitgestellte CFNTopicRuleProps, um die Standardwerte zu überschreiben
KinesisFirehoseProps?	`kinesisfirehose.CfnDeliveryStreamProps`	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für Kinesis Firehose Delivery Stream
Bestehend BuckeObj?	`s3.Bucket`	Vorhandene Instanz des S3 Bucket-Objekts, die sowohl diese als auch `bucketProps` führt zu einem -Fehler.
BucketProps?	`s3.BucketProps`	Der Benutzer hat Requisiten bereitgestellt, um die Standard-Requisiten für den S3-Bucket zu überschreiben. Wenn dies zur Verfügung gestellt wird, dann wird auch `bucketProps` ist ein Fehler.
LogGroupProps?	`logs.LogGroupProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardprops für die CloudWatch Logs-Protokollgruppe.

Muster-Eigenschaften

Name	Typ	Beschreibung
IoTactionSrole	iam.Role	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster für die IoT Regel erstellt wurde.
IottopicRule	iot.CfnTopicRule	Gibt eine Instanz der IoT -Themenregel zurück, die durch das Muster erstellt wurde.
KinesisFirehose	kinesisfirehose.CfnDeliveryStream	Gibt eine Instanz des Kinesis Firehose Übermittlungsdatenstroms zurück, der durch das Muster erstellt wurde.
KinesisFirehoselogGruppe	logs.LogGroup	Gibt eine Instanz der Protokollgruppe zurück, die durch das Muster erstellt wurde, an das Kinesis Firehose-Zugriffsprotokolle gesendet werden.
KinesisFirehoserole	iam.Role	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster für den Kinesis Firehose-Bereitstellungsstream erstellt wurde.
S3-Bucket?	s3.Bucket	Gibt eine Instanz des S3-Buckets zurück, der durch das Muster erstellt wurde.
s3loggingBucket?	s3.Bucket	Gibt eine Instanz des Protokollierungsbuckets zurück, der

Name	Typ	Beschreibung
		durch das Muster für den S3-Bucket erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon IoT Regel

- IAM-Rolle für den Zugriff auf die geringste Berechtigung für Amazon IoT konfigurieren

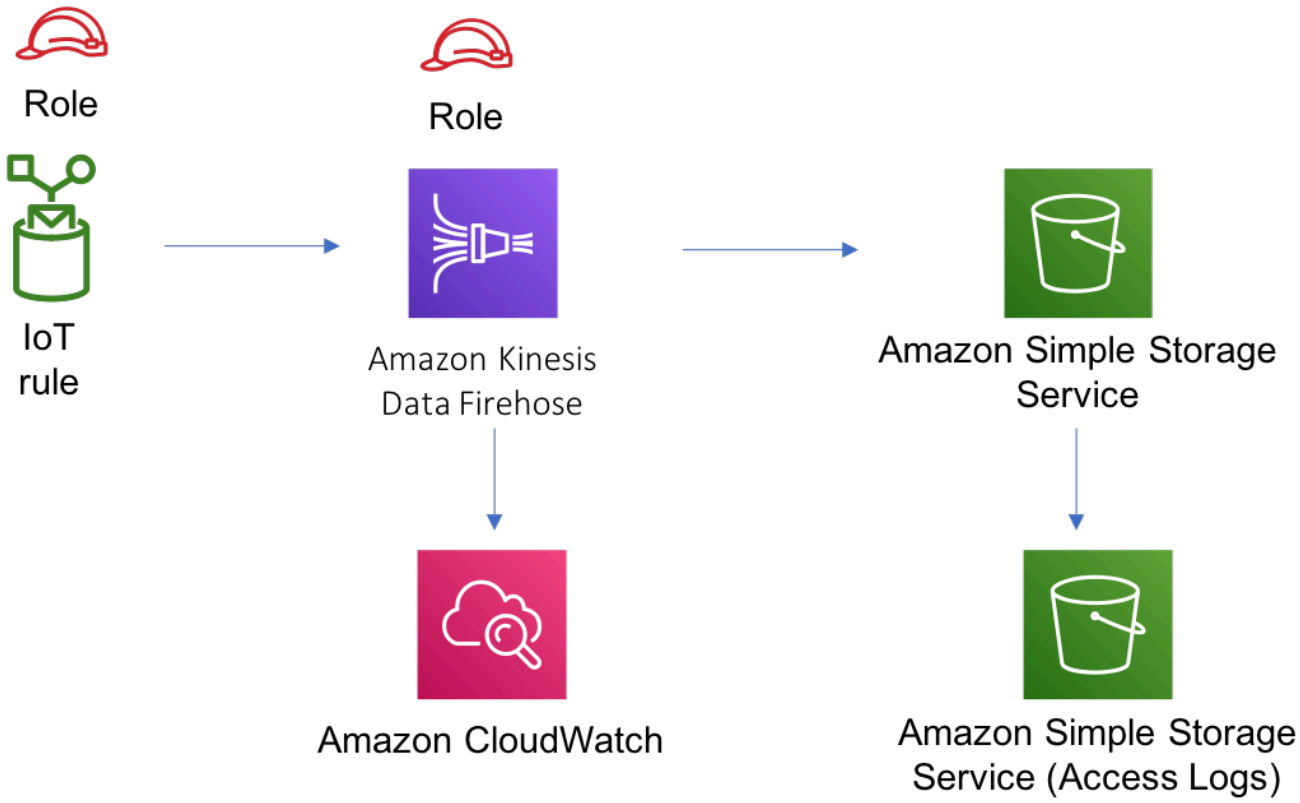
Amazon Kinesis Firehose

- Aktivieren der CloudWatch Protokollierung für Kinesis Firehose
- IAM-Rolle für den Zugriff auf die geringste Berechtigung für Amazon Kinesis Firehose konfigurieren

Amazon S3 Bucket

- Konfigurieren der Zugriffsprotokollierung für S3 Bucket
- Aktivieren der serverseitigen Verschlüsselung für S3 Bucket mit AWS verwaltetem KMS-Schlüssel
- Aktivieren der Versionierung für S3 Bucket
- Öffentlichen Zugriff für S3 Bucket nicht zulassen
- Beibehalten des S3-Bucket beim Löschen des CloudFormation -Stacks
- Wendet Lebenszyklusregel an, um nicht aktuelle Objektversionen nach 90 Tagen in Glacier-Speicher zu verschieben

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-konstrukte/aws-iot-kinesisfireschlauch-s3

aws-iot-lambda

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_iot_lambda`
Typoskript	`@aws-solutions-constructs/aws-iot-lambda`
Java	`software.amazon.awsconstructs.services.iotlambda`

Overview

Dieses AWS Solutions Constructs-Muster implementiert eine AWS IoT MQTT-Themenregel und ein AWS Lambda Funktionsmuster.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`IotToLambdaProps`

Muster-Konstrukt

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird bereitgestellt.

Name	Typ	Beschreibung
IottopicRuleProps?	`iot.CfnTopicRulePr ops`	Vom Benutzer bereitgestellte CFNTopicRuleProps, um die Standardwerte zu überschre iben

Muster-Eigenschaften

Name	Typ	Beschreibung
IottopicRule	`iot.CfnTopicRule`	Gibt eine Instanz der IoT - Themenregel zurück, die durch das Muster erstellt wurde.
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon IoT Regel

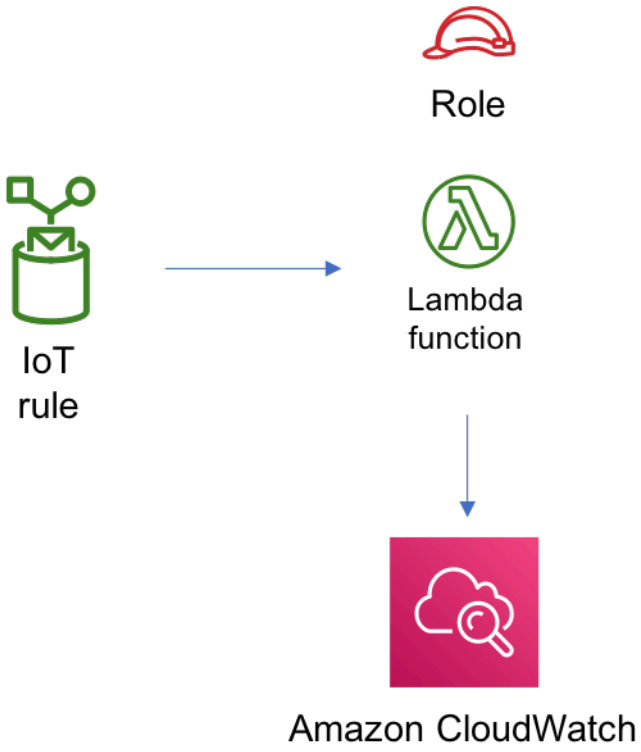
- Konfigurieren Sie die IAM-Rolle für den Zugriff auf die geringste Berechtigung für Amazon IoT.

AWS Lambda-Funktion

- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- Aktivieren Sie die X-Ray Nachverfolgung.

- Festlegen von Umgebungsvariablen:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-iot-lambda

aws-iot-lambda-dynamodb

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische

VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_iot_lambda_dynamodb`
Typoskript	`@aws-solutions-constructs/aws-iot-lambda-dynamodb`
Java	`software.amazon.awsconstructs.services.iotlambdadynamodb`

Overview

Dieses AWS Solutions Constructs-Muster implementiert eine AWS IoT -Themenregel, eine AWS Lambda Funktion und Amazon DynamoDB -Tabelle mit den geringsten Berechtigungen.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`IotToLambdaToDynamoDBProps`

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion

Name	Typ	Beschreibung
		zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird bereitgestellt.
IottopicRuleProps	`iot.CfnTopicRuleProps`	Vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten
DynamoTableProps?	`dynamodb.TableProps`	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für DynamoDB -Tabelle
TabelleBerechtigungen?	`string`	Optionale Tabellenberechtigungen, die der Lambda Funktion erteilt werden sollen. Es können folgende Optionen angegeben werden:`All`,`Read`,`ReadWrite`, oder`Write`.

Muster-Eigenschaften

Name	Typ	Beschreibung
DynamoTable	`dynamodb.Table`	Gibt eine Instanz der DynamoDB -Tabelle zurück, die durch das Muster erstellt wurde.
IottopicRule	`iot.CfnTopicRule`	Gibt eine Instanz der IoT -Themenregel zurück, die durch das Muster erstellt wurde.

Name	Typ	Beschreibung
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon IoT Regel

- Konfigurieren Sie die IAM-Rolle für den Zugriff auf die geringste Berechtigung für Amazon IoT.

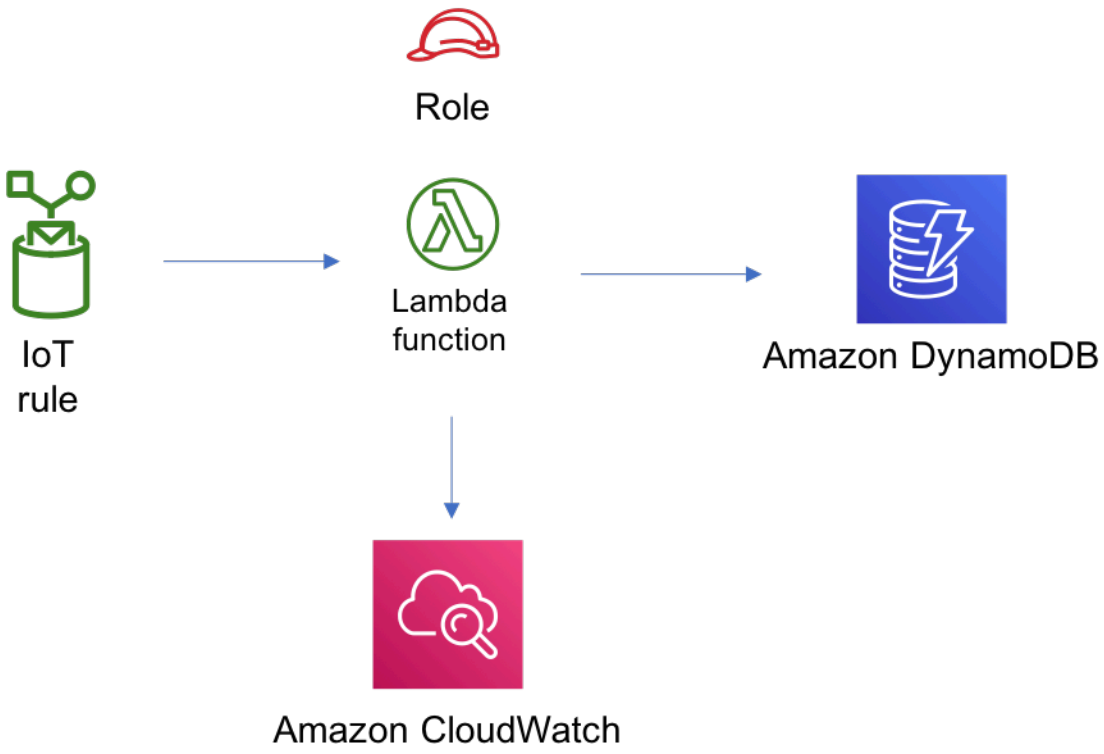
AWS Lambda-Funktion

- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- Aktivieren Sie X-Ray blaufverfolgung.
- Festlegen von Umgebungsvariablen:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Amazon DynamoDB Tabelle

- Legen Sie den Abrechnungsmodus für DynamoDB -Tabelle auf On-Demand fest (Bezahlen pro Anforderung).
- Aktivieren Sie die serverseitige Verschlüsselung für DynamoDB Table mithilfe von AWS verwaltetem KMS-Schlüssel.
- Erstellt einen Partitionsschlüssel namens 'id' für DynamoDB -Tabelle.
- Behalten Sie die Tabelle beim Löschen des CloudFormation-Stacks bei.
- Ermöglicht kontinuierliche Sicherungen und zeitpunktbezogene Wiederherstellungen.

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-konstrukte/aws-iot-lambda-dynamodb

aws-kinesisfireschlauch-s3

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

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

Overview

Dieses AWS Solutions Construct implementiert einen Amazon Kinesis Data Firehose Delivery-Stream, der mit einem Amazon S3 Bucket verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`

- `id`string
- propsKinesisFirehoseToS3Props

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
BucketProps?	s3.BucketProps	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für den S3-Bucket zu überschre iben.
Bestehend BuckeObj?	s3.IBucket	Optionale vorhandene Instanz von S3 Bucket. Wenn dies zur Verfügung gestellt wird, dann wird auch die Bereitste llung von`bucketProps` ist ein Fehler.
ExistingLoggingBuckeObj?	s3.IBucket	Optionale vorhandene Instanz der Protokollierung von S3 Bucket für den S3-Bucket, der durch das Muster erstellt wurde.
KinesisFirehoseProps?	kinesisfirehose.CfnDeliveryStreamProps | any	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für Kinesis Firehose Delivery Stream zu überschreiben.
LogGroupProps?	logs.LogGroupProps	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für die CloudWatchLogs Group zu überschreiben.

Muster-Eigenschaften

Name	Typ	Beschreibung
KinesisFirehose	kinesisfirehose.CfnDeliveryStream	Gibt eine Instanz von KinesisFireHose.cfnDelivery Stream zurück, die vom Konstrukt erstellt wurde.
KinesisFirehoselogGruppe	logs.LogGroup	Gibt eine Instanz der Logs.logGroup zurück, die vom Konstrukt für Kinesis Data Firehose Delivery Stream erstellt wurde.
KinesisFirehoserole	iam.Role	Gibt eine Instanz des IAM.role zurück, die vom Konstrukt für Kinesis Data Firehose Delivery Stream erstellt wurde.
S3Eimer?	s3.Bucket	Gibt eine Instanz von S3.Bucket zurück, die vom Konstrukt erstellt wurde.
s3loggingBucket?	s3.Bucket	Gibt eine Instanz von S3.Bucket zurück, die vom Konstrukt als Protokoll ierungsbucket für den primären Bucket erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

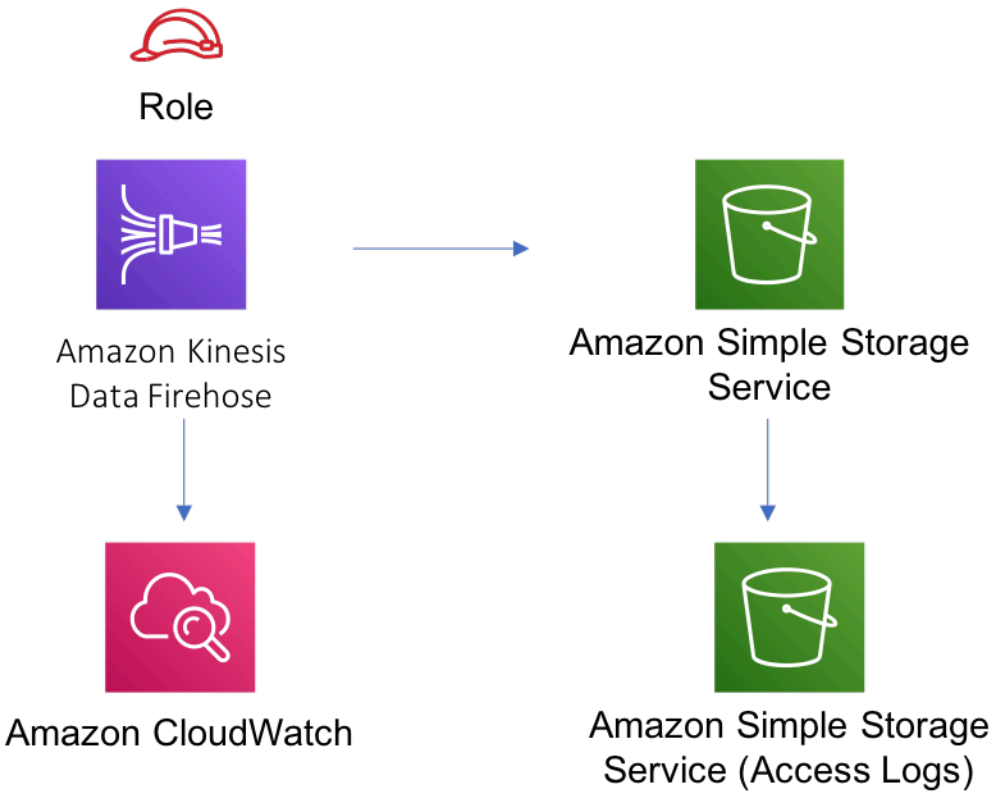
Amazon Kinesis Firehose

- Aktivieren der CloudWatch Protokollierung für Kinesis Firehose
- IAM-Rolle für den Zugriff auf die geringste Berechtigung für Amazon Kinesis Firehose konfigurieren

Amazon S3 Bucket

- Konfigurieren der Zugriffsprotokollierung für S3 Bucket
- Aktivieren der serverseitigen Verschlüsselung für S3 Bucket mit AWS verwaltetem KMS-Schlüssel
- Aktivieren der Versionierung für S3 Bucket
- Öffentlichen Zugriff für S3 Bucket nicht zulassen
- Beibehalten des S3-Bucket beim Löschen des CloudFormation -Stacks
- Erzwingen der Verschlüsselung von Daten während der Übertragung
- Wendet Lebenszyklusregel an, um nicht aktuelle Objektversionen nach 90 Tagen in Glacier-Speicher zu verschieben

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:	
	@aws -solutions-konstrukte/aws-kinesisfireschlauch-s3

aws-kinesisfireschlauch-s3-und-kinesisanalytics

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantic VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_kinesisfirehose_s3_and_kinesisanalytics`
TypeScript	`@aws-solutions-constructs/aws-kinesisfirehose-s3-and-kinesisanalytics`
Java	`software.amazon.awsconstructs.services.kinesisfirehoses3kinesisanalytics`

Overview

Dieses AWS Solutions Construct implementiert einen Amazon Kinesis Firehose Bereitstellungsstream, der mit einem Amazon S3 Bucket und einer Amazon Kinesis Analytics Anwendung verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsKinesisFirehoseToAnalyticsAndS3Props

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
KinesisFirehoseProps?	kinesisFirehose.CfnDeliveryStreamProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für den Kinesis Firehose-Bereitstellungsstream.
KinesisAnalyticsProps?	kinesisAnalytics.CfnApplicationProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für die Kinesis Analytics-Anwendung.
Bestehend BuckeObj?	s3.IBucket	Vorhandene Instanz des S3 Bucket-Objekts. Wenn dies vorgesehen ist, dann wird auch die Bereitstellung vonbucketProps ist ein Fehler. |

Name	Typ	Beschreibung
BucketProps?	s3.BucketProps	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für den Bucket zu überschreiben. Wird ignoriert, wenn ein existingBucketObj wird bereitgestellt.
LogGroupProps?	logs.LogGroupProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardprops für die CloudWatch Logs-Protokollgruppe

Muster-Eigenschaften

Name	Typ	Beschreibung
KinesisAnalytics	kinesisAnalytics.CfnApplication	Gibt eine Instanz der Kinesis Analytics-Anwendung zurück, die durch das Muster erstellt wurde.
KinesisFirehose	kinesisfirehose.CfnDeliveryStream	Gibt eine Instanz des Kinesis Firehose-Lieferdatenstroms zurück, der durch das Muster erstellt wurde.
KinesisFirehoselogGruppe	logs.LogGroup	Gibt eine Instanz der Protokollgruppe zurück, die durch das Muster erstellt wurde, an das Kinesis Firehose-Zugriffsprotokolle gesendet werden.

Name	Typ	Beschreibung
KinesisFirehoserole	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster für den Kinesis Firehose-Bereitstellungsstr eam erstellt wurde.
S3Eimer?	`s3.Bucket`	Gibt eine Instanz des S3-Buckets zurück, der durch das Muster erstellt wurde.
s3loggingBucket?	`s3.Bucket`	Gibt eine Instanz des Protokoll ierungsbuckets zurück, der durch das Muster für den S3-Bucket erstellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon Kinesis Firehose

- Aktivieren der CloudWatch Protokollierung für Kinesis Firehose
- IAM-Rolle für den Zugriff auf die geringste Berechtigung für Amazon Kinesis Firehose konfigurieren

Amazon S3 Bucket

- Konfigurieren der Zugriffsprotokollierung für S3 Bucket
- Aktivieren der serverseitigen Verschlüsselung für S3 Bucket mit AWS verwaltetem KMS-Schlüssel
- Aktivieren der Versionierung für S3 Bucket
- Öffentlichen Zugriff für S3 Bucket nicht zulassen
- Beibehalten des S3-Bucket beim Löschen des CloudFormation -Stacks
- Erzwingen der Verschlüsselung von Daten während der Übertragung
- Wendet Lebenszyklusregel an, um nicht aktuelle Objektversionen nach 90 Tagen in Glacier-Speicher zu verschieben

Amazon Kinesis Data Analytics

- Konfigurieren der IAM-Rolle für den Zugriff auf die geringste Berechtigung für Amazon Kinesis Analytics

Architecture

GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-konstrukte/aws-kinesisfireschlauch-s3-und-kinesisanalytics

aws-kinesisstreams-gluejob

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_kinesis_streams_gluejob`
Typoskript	`@aws-solutions-constructs/aws-kinesisstreams-gluejob`
Java	`software.amazon.awsconstructs.services.kinesisstreamsgluejob`

Overview

Dieses AWS Solutions Construct stellt einen Amazon Kinesis Data Stream bereit und konfiguriert einen AWS Glue e-Job für die Durchführung einer benutzerdefinierten ETL-Transformation mit den entsprechenden Ressourcen/Eigenschaften für Interaktion und Sicherheit. Außerdem wird ein Amazon S3 Bucket erstellt, in dem das Python-Skript für den AWS Glue e-Job hochgeladen werden kann.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsKinesisstreamsToGluejobProps

Musterkonstrukt Requisiten

Name	Typ	Beschreibung
KinesisStreamProps?	kinesis.StreamProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für den Amazon Kinesis Datenstrom.
Bestehend StreamObj?	kinesis.Stream	Vorhandene Instanz von Kinesis Stream, die sowohl diese als auchkinesisStreamProps Ein Fehler führt zu einem.
GluejobProps?	cfnJob.CfnJobProps	Vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für den AWS Glue e-Auftrag zu überschreiben.
ExistingGlueJob?	cfnJob.CfnJob	Vorhandene Instanz von AWS Glue Job, die sowohl diese als auchglueJobProps Ein Fehler führt zu einem.
Bestehende Datenbank?	CfnDatabase	Vorhandene AWS Glue e-Datenbank, die mit diesem Konstrukt verwendet werden soll. Wenn dies gesetzt ist,

Name	Typ	Beschreibung
		danndatabaseProps wird ignoriert.
DatabaseProps?	CfnDatabaseProps	Vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten, die zum Erstellen der AWS Glue e-Datenbank verwendet werden.
ExistingTable?	CfnTable	Vorhandene Instanz der AWS Glue e-Tabelle. Wenn dies gesetzt ist, danntableProps undfieldSchema werden ignoriert.
TableProps?	CfnTableProps	Vom Benutzer bereitgestellte Requisiten zum Überschreiben von Standardrequisiten, die zum Erstellen einer AWS Glue e-Tabelle verwendet werden.
FieldSchema?	CfnTable.ColumnProperty[]	Vom Benutzer bereitgestellte Schemastruktur zum Erstellen einer AWS Glue e-Tabelle
OutputDataStore?	SinkDataStoreProps	Vom Benutzer bereitgestellte Requisiten für einen Amazon S3 Bucket, der die Ausgabe des AWS Glue e-Auftrags speichert. Derzeit unterstützt Amazon S3 nur als Ausgabe-Datenspeichertyp.

SinkDataStoreProps

Name	Typ	Beschreibung
Existings3OutputBucket?	`Bucket`	Eine vorhandene Instanz von S3-Bucket, in dem die Daten geschrieben werden sollen. Bereitstellung sowohl dieser als auch`outputBucketProps` Ein Fehler führt zu einem.
OutputBucketProps	`BucketProps`	Vom Benutzer bereitgestellte Bucket-Eigenschaften, um den Amazon S3 Bucket zu erstellen, der zum Speichern der Ausgabe des AWS Glue e-Auftrags verwendet wird.
DataStoreType	`SinkStoreType`	Data-Store-Typ versenken.

SinkStoreType

Aufzählung von Datenspeichertypen, die S3, DynamoDB, DocumentDB, RDS oder Redshift umfassen können. Die aktuelle Konstruktimplementierung unterstützt nur S3, kann aber in Zukunft andere Ausgabetypen hinzufügen.

Name	Typ	Beschreibung
S3	`string`	S3-Speichertyp

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon Kinesis Stream

- Konfigurieren Sie die IAM-Rolle für den Zugriff auf die geringste Berechtigung für den Amazon Kinesis Datenstrom.
- Aktivieren Sie die serverseitige Verschlüsselung für den Amazon Kinesis Stream mit einem AWS Managed KMS-Schlüssel.
- Stellen Sie bewährte Amazon CloudWatch Alarme für den Amazon Kinesis Stream bereit.

Glue e-Job

- Erstellen Sie eine AWS Glue e-Sicherheitskonfiguration, die die Verschlüsselung für CloudWatch, Job Bookmarks und S3 konfiguriert. CloudWatch ch- und Job -Lesezeichen werden mit AWS Managed KMS Key verschlüsselt, der für AWS Glue Service erstellt wurde. Der S3-Bucket ist mit dem SSE-S3-Verschlüsselungsmodus konfiguriert.
- Konfigurieren Sie Service-Rollenrichtlinien, mit denen AWS Glue von Amazon Kinesis Data Streams gelesen werden kann.

Glue e-Datenbank

- Erstellen Sie eine AWS Glue e-Datenbank. Zur Datenbank wird eine AWS Glue e-Tabelle hinzugefügt. In dieser Tabelle wird das Schema für die Datensätze definiert, die im Amazon Kinesis Datenstrom gepuffert werden.

Glue e-Tabelle

- Erstellen Sie eine AWS Glue e-Tabelle. Die Tabellenschemadefinition basiert auf der JSON-Struktur der Datensätze, die im Amazon Kinesis Data Stream gepuffert werden.

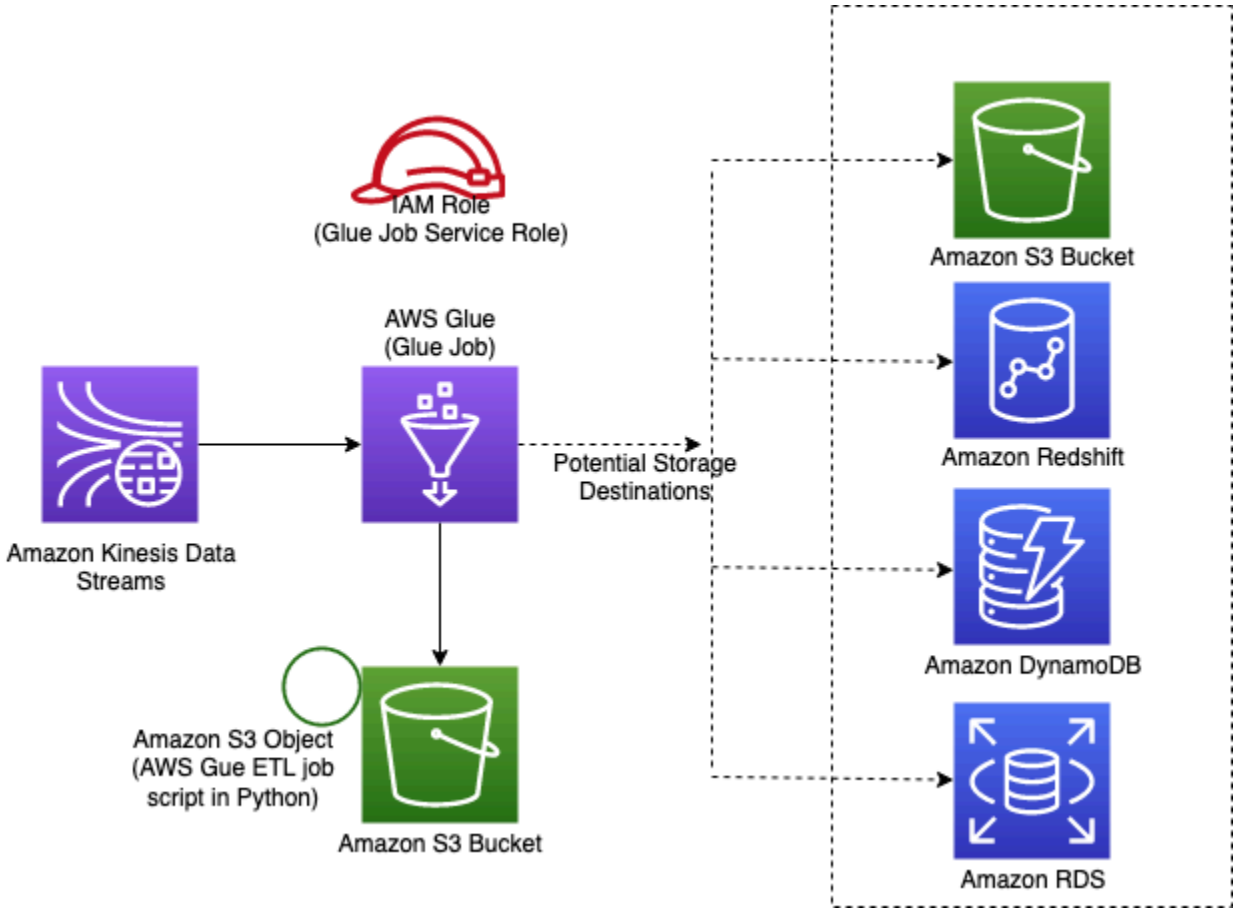
IAM Role (IAM-Rolle)

- Eine Rolle zur Auftragsausführung, die Privilegien hat, 1) das ETL-Skript vom Amazon S3 Bucket-Speicherort zu lesen, 2) Datensätze aus dem Amazon Kinesis Datenstrom zu lesen und 3) den Amazon Glue e-Auftrag auszuführen.

Ausgang S3-Bucket

- Ein Amazon S3 Bucket, in dem die Ausgabe der ETL-Transformation gespeichert werden soll. Dieser Bucket wird als Argument an den erstellten AWS Glue e-Auftrag übergeben, sodass er im ETL-Skript zum Schreiben von Daten verwendet werden kann.

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-konstrukte/aws-kinesisstreams-gluejob

aws-kinesisstreams-kinesisfireschlauch-s3

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_kinesisstreams_kinesisfirehose_s3`
Typoskript	`@aws-solutions-constructs/aws-kinesis-streams-kinesis-firehose-s3`
Java	`software.amazon.awsconstructs.services.kinesisstreamskinesisfirehoses3`

Overview

Dieses AWS Solutions Construct implementiert einen Amazon Kinesis Data Stream (KDS), der mit Amazon Kinesis Data Firehose (KDF) -Bereitstellungsstrom verbunden ist, der mit einem Amazon S3 Bucket verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsKinesisStreams...ToS3Props

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
BucketProps?	s3.BucketProps	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für den S3-Bucket zu überschre iben.
CreateCloudWatchalarms?	boolean	Optional, ob empfohlene CloudWatch Alarme erstellt werden sollen.
Bestehend BuckeObj?	s3.IBucket	Optionale vorhandene Instanz des S3 Bucket-Objekts. Wenn dies zur Verfügung gestellt wird, dann wird auchbucketProps ist ein Fehler.
ExistingLoggingBuckeObj?	s3.IBucket	Optionale vorhandene Instanz der Protokollierung des S3

Name	Typ	Beschreibung
		Bucket-Objekts für den S3-Bucket, der durch das Muster erstellt wurde.
Bestehend StreamObj?	kinesis.Stream	Vorhandene Instanz von Kinesis Stream, die sowohl diese als auch `kinesisStreamProps` führt zu einem -Fehler.
KinesisFirehoseProps?	aws-kinesisfirehose.CfnDeliveryStreamProps | any	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für Kinesis Firehose Delivery Stream zu überschreiben.
KinesisStreamProps?	kinesis.StreamProps	Optionale vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für den Kinesis Stream zu überschreiben.
LogGroupProps?	logs.LogGroupProps	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für die CloudWatchLogs s-Protokollgruppe zu überschreiben.

Muster-Eigenschaften

Name	Typ	Beschreibung
CloudwatChalarms?	cloudwatch.Alarm[]	Gibt eine Liste der CloudWatch.Alarm-Instanzen zurück, die vom Konstrukt erstellt wurden.
KinesisFirehose	kinesisfirehose.CfnDeliveryStream	Gibt eine Instanz von KinesisFireHose.cfnDeliveryStream zurück, die vom Konstrukt erstellt wurde.
KinesisFirehoselogGruppe	logs.LogGroup	Gibt eine Instanz der Logs.logGroup zurück, die vom Konstrukt für Kinesis Data Firehose Delivery Stream erstellt wurde.
KinesisFirehoserole	iam.Role	Gibt eine Instanz des IAM.role zurück, die vom Konstrukt für Kinesis Data Firehose Delivery Stream erstellt wurde.
KinesisStreamRole	iam.Role	Gibt eine Instanz der IAM.role zurück, die vom Konstrukt für Kinesis Stream erstellt wurde.
S3-Bucket?	s3.Bucket	Gibt eine Instanz von S3.Bucket zurück, die vom Konstrukt erstellt wurde.
s3loggingBucket?	s3.Bucket	Gibt eine Instanz von S3.Bucket zurück, die vom Konstrukt als Protokollierungsbucket für den primären Bucket erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon Kinesis Stream

- Konfigurieren der IAM-Rolle für den Zugriff auf die geringste Berechtigung für Kinesis Stream
- Aktivieren der serverseitigen Verschlüsselung für Kinesis Stream mit AWS Managed KMS Key
- Bereitstellen von Best Practices CloudWatch Alarmen für den Kinesis Stream

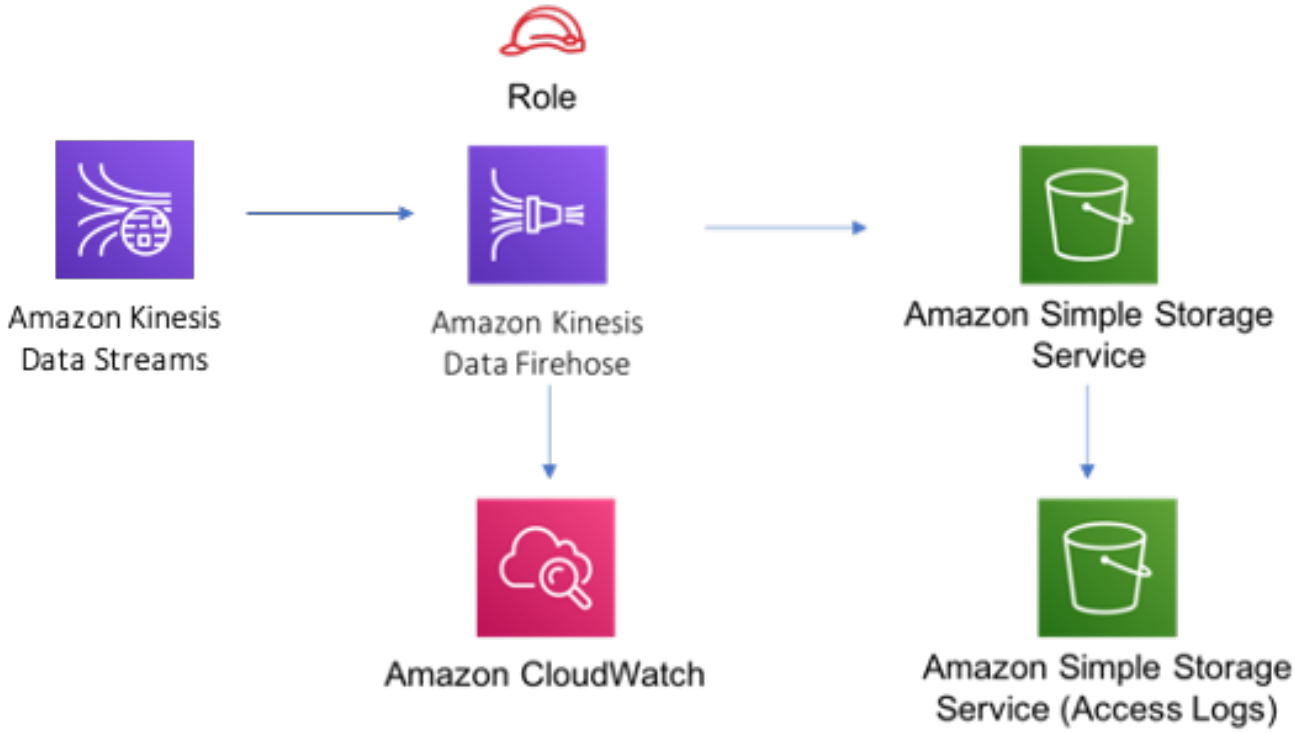
Amazon Kinesis Firehose

- Aktivieren der CloudWatch Protokollierung für Kinesis Firehose
- IAM-Rolle für den Zugriff auf die geringste Berechtigung für Amazon Kinesis Firehose konfigurieren

Amazon S3 Bucket

- Konfigurieren der Zugriffsprotokollierung für S3-Bucket
- Aktivieren der serverseitigen Verschlüsselung für S3-Bucket mit AWS verwaltetem KMS-Schlüssel
- Erzwingen der Verschlüsselung von Daten während der Übertragung
- Bucket-Versionierung aktivieren
- Öffentlichen Zugriff für S3-Bucket nicht zulassen
- Beibehalten des S3-Bucket beim Löschen des CloudFormation -Stacks
- Lebenszyklusregel anwenden, um nicht aktuelle Objektversionen nach 90 Tagen in Glacier-Speicher zu verschieben

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-kinesisstreams-kinesisfireschlauch-s3

aws-kinesisstreams-lambda

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws-kinesis-streams-lambda`
Typoskript	`@aws-solutions-constructs/aws-kinesisstreams-lambda`
Java	`software.amazon.awsconstructs.services.kinesisstreamslambda`

Overview

Dieses AWS Solutions Construct stellt eine Kinesis Stream- und Lambda Funktion mit den entsprechenden Ressourcen/Eigenschaften für Interaktion und Sicherheit bereit.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsKinesisStreamsToLambdaProps

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
Gibt es LambdaObj?	lambda.Function	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auchlambdaFunctionProps führt zu einem -Fehler.
LambdaFunctionProps?	lambda.FunctionProps	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn einexistingLambdaObj wird bereitgestellt.
KinesisStreamProps?	kinesis.StreamProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-

Name	Typ	Beschreibung
		Requisiten für den Kinesis Stream.
Bestehend StreamObj?	`kinesis.Stream`	Vorhandene Instanz von Kinesis Stream, die sowohl diese als auch`kinesisStreamProps` führt zu einem -Fehler.
KinesisEventSourceProps?	`aws-lambda-event-sources.KinesisEventSourceProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für die Lambda Ereignisquellenzuordnung.
CreateCloudWatchalarms	`boolean`	Gibt an, ob empfohlene CloudWatch Alarme erstellt werden sollen.

Muster-Eigenschaften

Name	Typ	Beschreibung
KinesisisStream	`kinesis.Stream`	Gibt eine Instanz des Kinesis Streams zurück, der durch das Muster erstellt wurde.
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
KinesisStreamRole	`iam.Role`	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster für den Kinesis Stream erstellt wurde.

Name	Typ	Beschreibung
CloudwatChalarms?	`cloudwatch.Alarm[]`	Gibt eine Liste eines oder mehrerer CloudWatch Alarme zurück, die durch das Muster erstellt wurden.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

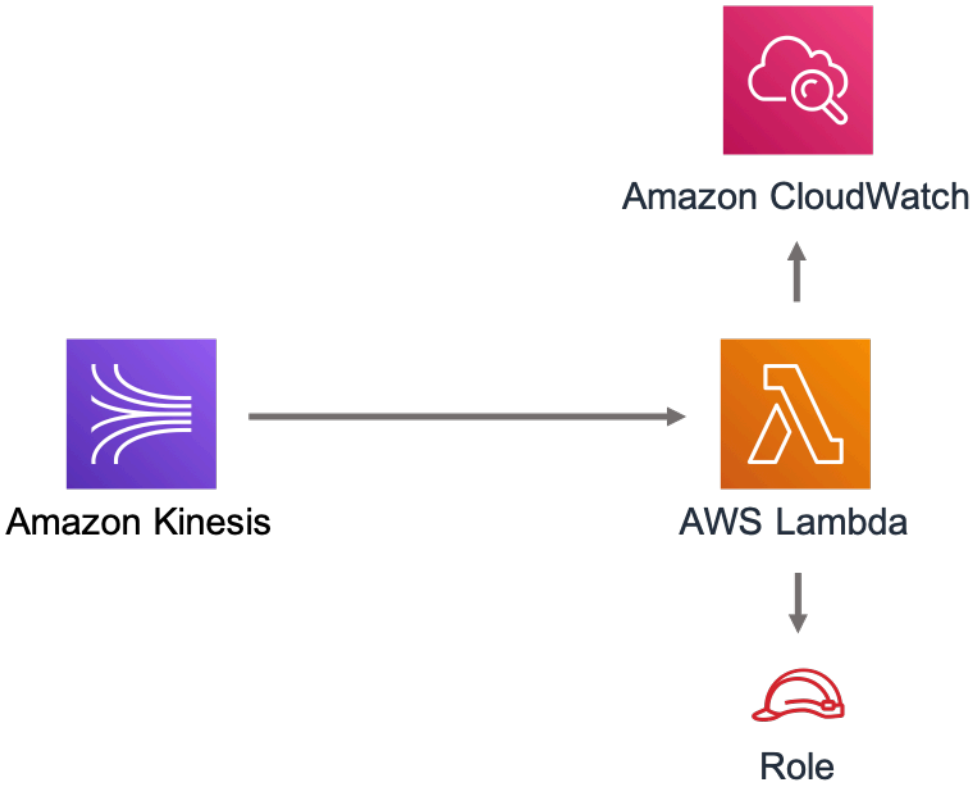
Amazon Kinesis Streams

- Konfigurieren Sie die IAM-Rolle für den Zugriff auf die geringste Berechtigung für Kinesis Stream.
- Aktivieren Sie die serverseitige Verschlüsselung für Kinesis Stream mit AWS Managed KMS Key.
- Stellen Sie bewährte CloudWatch Alarme für den Kinesis Stream bereit.

AWS Lambda-Funktion

- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- Aktivieren Sie die X-Ray Nachverfolgung.
- Funktionen zur Fehlerbehandlung aktivieren: Bisect bei Funktion Fehler aktivieren; Standardeinstellung Maximales Datensatzalter (24 Stunden) festlegen; Standardeinstellung Maximale Wiederholungsversuche (500) festlegen und SQS-Warteschlange für Dead-Letter als Ziel bei einem Fehler bereitstellen.
- Festlegen von Umgebungsvariablen:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-kinesisstreams-lambda

aws-lambda-dynamodb

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht denSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_lambda_dynamodb`
TypeScript	`@aws-solutions-constructs/aws-lambda-dynamodb`
Java	`software.amazon.awsconstructs.services.lambdadynamodb`

Overview

Dieses AWS Solutions Construct implementiert die AWS Lambda Funktion und die Amazon DynamoDB -Tabelle mit den geringsten Berechtigungen.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`LambdaToDynamoDBProps`

Muster-Konstrukt

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird bereitgestellt.
DynamoTableProps?	`dynamodb.TableProps`	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für DynamoDB -Tabelle

Name	Typ	Beschreibung
ExistingTableObj?	dynamodb.Table	Vorhandene Instanz des DynamoDB -Tabellen objekts, die sowohl diese als auch `dynamoTableProps` führt zu einem - Fehler.
TabelleBerechtigungen?	string	Optionale Tabellenberechtigungen, die der Lambda Funktion erteilt werden sollen. Sie können eine der folgenden Optionen angeben: `All`, `Read`, `ReadWrite`, oder `Write`.
TableEnvironmentVariableName?	`string`	Optionaler Name für die DynamoDB -Tabellen umgebungsvariable, die für die Lambda Funktion festgelegt wurde.

Name	Typ	Beschreibung
ExistenzVPC?	`ec2.IVpc`	Eine optionale vorhandene VPC, in der dieses Pattern bereitgestellt werden soll. Bei der Bereitstellung in einer VPC verwendet die Lambda Funktion ENIs in der VPC für den Zugriff auf Netzwerkressourcen, und ein Gateway-Endpunkt wird in der VPC für Amazon DynamoDB erstellt. Wenn eine vorhandene VPC bereitgestellt wird, wird die`deployVpc` kann nicht`true`. Diese Verwendung von`ec2.IVpc`, damit Clients VPCs bereitstellen können, die außerhalb des Stacks vorhanden sind, mithilfe des`ec2.Vpc.fromLookup()` -Methode.

Name	Typ	Beschreibung
VPCProps?	`ec2.VpcProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die neue VPC zu überschreiben.`enableDnsHostnames` ,`enableDnsSupport` ,`natGateways` , und`subnetConfiguration` werden durch das Muster festgelegt, so dass alle Werte für die hier angegebenen Eigenschaften überschrieben werden. Wenn`deployVpc` ist nicht`true`wird diese Eigenschaft ignoriert.

Name	Typ	Beschreibung
DeployVPC?	`boolean`	So erstellen Sie eine neue VPC`vpcProps`, in das dieses Muster bereitgestellt werden soll. Wenn Sie dies auf true setzen, wird die minimale, privatste VPC bereitgestellt, um das Pattern auszuführen: • Ein isoliertes Subnetz in jeder Availability Zone, die vom CDK-Programm verwendet wird • `enableDnsHostnames` und`enableDnsSupport` werden beide auf`true` Wenn diese Eigenschaft`true`, dann`existingVpc` kann nicht angegeben werden. Standardeinstellung: `false`.

Muster-Eigenschaften

Name	Typ	Beschreibung
DynamoTable	`dynamodb.Table`	Gibt eine Instanz der DynamoDB -Tabelle zurück, die durch das Muster erstellt wurde.

Name	Typ	Beschreibung
LambdaFUNTION	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
vpc?	`ec2.IVpc`	Gibt eine Schnittstelle auf der VPC zurück, die vom Muster verwendet wird (falls vorhanden). Dies kann eine VPC sein, die durch das Muster oder die VPC erstellt wurde, die dem Musterkon struktor zur Verfügung gestellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

AWS Lambda-Funktion

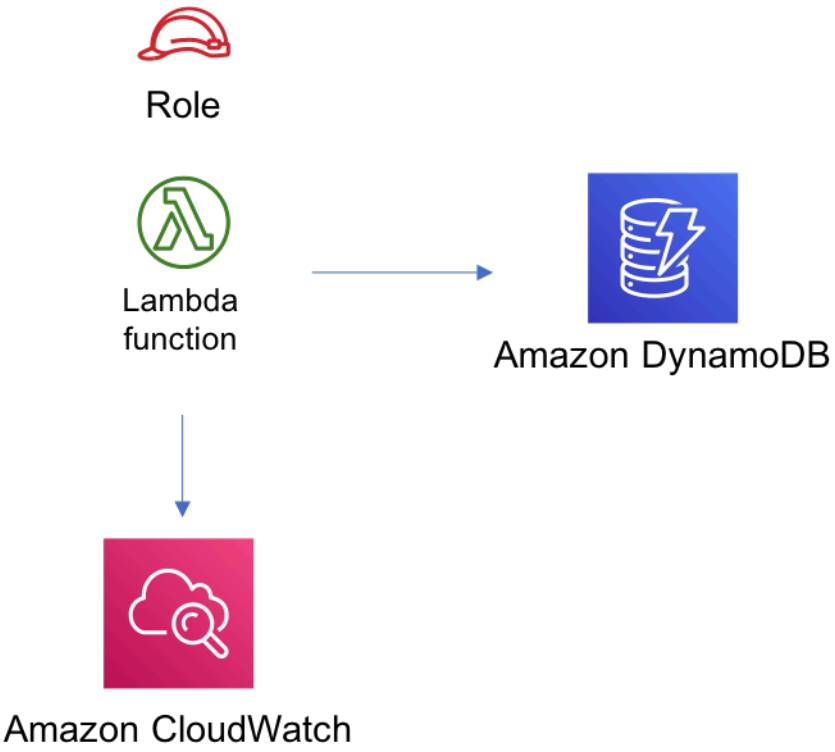
- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- Aktivieren von X-Ray blaufverfolgung.
- Festlegen von Umgebungsvariablen:
  - `DDB_TABLE_NAME` (Standard)
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Amazon DynamoDB Tabelle

- Legen Sie den Abrechnungsmodus für DynamoDB -Tabelle auf On-Demand fest (Bezahlen pro Anforderung).

- Aktivieren Sie die serverseitige Verschlüsselung für DynamoDB Table mithilfe von AWS verwaltetem KMS-Schlüssel.
- Erstellt einen Partitionsschlüssel namens 'id' für DynamoDB -Tabelle.
- Behalten Sie die Tabelle beim Löschen des CloudFormation-Stacks bei.
- Ermöglicht kontinuierliche Sicherungen und zeitpunktbezogene Wiederherstellungen.

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-konstrukte/aws-lambda-dynamodb

aws-lambda-elasticsearch-kibana

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_lambda_elasticsearch_kibana`
Typoskript	`@aws-solutions-constructs/aws-lambda-elasticsearch-kibana`
Java	`software.amazon.awsconstructs.services.lambdaelasticsearchkibana`

Overview

Dieses AWS Solutions Construct implementiert eine AWS Lambda Funktion und eine Amazon Elasticsearch Service Domäne mit den am wenigsten privilegierten Berechtigungen.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`LambdaToElasticSearchAndKibanaProps`

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaf

Name	Typ	Beschreibung
		ten für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird bereitgestellt.
ESDomainProps?	`elasticsearch.CfnDomainProps`	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für den Amazon Elasticsearch Service
domainName	`string`	Domänenname für den Cognito und den Amazon Elasticsearch Service
CognitoDomainName?	`string`	Optionaler Cognito Domänennamen. Wenn angegeben, wird es für die Cognito Domain verwendet , und`domainName` wird für die Elasticsearch-Domain verwendet.
CreateCloudWatchalarms	`boolean`	Ob empfohlene CloudWatch Alarme erstellt werden sollen.
DomainEndpointEnvironmentVariableName?	`string`	Optionaler Name für die ElasticSearch Domänenendpunkt-Umgebungsvariable, die für die Lambda Funktion festgelegt wurde.

Muster-Eigenschaften

Name	Typ	Beschreibung
Cloudwatchalarms?	cloudwatch.Alarm[]	Gibt eine Liste eines oder mehrerer CloudWatch Alarme zurück, die durch das Muster erstellt wurden.
ElasticSearchDomain	elasticsearch.CfnDomain	Gibt eine Instanz der Elasticsearch-Domäne zurück, die durch das Muster erstellt wurde.
ElasticSearchDomainRole	iam.Role	Gibt eine Instanz der IAM-Rolle zurück, die durch das Muster für die Elasticsearch-Domäne erstellt wurde.
IdentityPool	cognito.CfnIdentityPool	Gibt eine Instanz des Cognito Identitätspools zurück, der durch das Muster erstellt wurde.
LambdaFunction	lambda.Function	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
userPool	cognito.UserPool	Gibt eine Instanz des Cognito Benutzerpools zurück, die durch das Muster erstellt wurde.
UserPoolClient	cognito.UserPoolClient	Gibt eine Instanz des Cognito Benutzerpool-Clients zurück, die durch das Muster erstellt wurde.

Lambda-Funktion

Dieses Muster erfordert eine Lambda Funktion, die Daten aus dem DynamoDB Stream in den Elasticsearch-Dienst posten kann. Eine Beispielfunktion wird bereitgestelltHier gilt:.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

AWS Lambda-Funktion

- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- X-Ray Nachverfolgung aktivieren.
- Festlegen von Umgebungsvariablen:
  - `DOMAIN_ENDPOINT` (Standard)
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

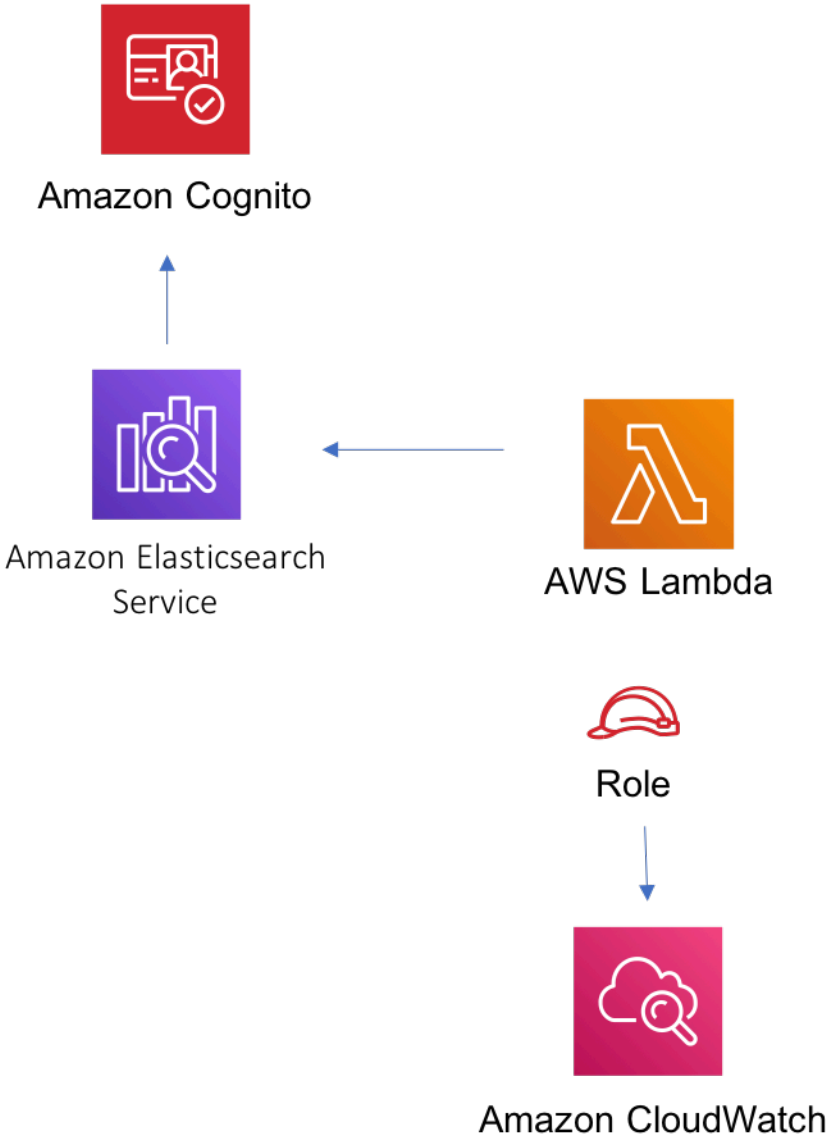
Amazon Cognito

- Festlegen der Kennwortrichtlinie für Benutzerpools.
- Erzwingen Sie den erweiterten Sicherheitsmodus für Benutzerpools.

Amazon Elasticsearch Service

- Stellen Sie Best Practices CloudWatch Alarme für die Elasticsearch-Domäne bereit.
- Sichern Sie den Zugriff auf das Kibana Dashboard mit Cognito User Pools.
- Aktivieren Sie die serverseitige Verschlüsselung für Elasticsearch-Domäne mithilfe von AWS verwaltetem KMS-Schlüssel.
- Aktivieren Sie die Knoten-zu-Knoten-Verschlüsselung für Elasticsearch-Domäne.
- Konfigurieren Sie den Cluster für die Amazon ES-Domäne.

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-konstrukte/aws-lambda-elasticsearch-kibana

aws-lambda-S3

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantic Versionierung-Modell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_lambda_s3`
Typoskript	`@aws-solutions-constructs/aws-lambda-s3`
Java	`software.amazon.awsconstructs.services.lambdas3`

Overview

Diese AWS Solutions Construct implementiert eine AWS Lambda -Funktion, die mit einem Amazon S3 Bucket verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`LambdaToS3Props`

Muster-Konstrukt

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird bereitgestellt.
Bestehend BuckeObj?	`s3.IBucket`	Vorhandene Instanz des S3 Bucket-Objekts. Wenn dies

Name	Typ	Beschreibung
		zur Verfügung gestellt wird, dann wird auch die Bereitstellung `bucketProps` ist ein Fehler.
BucketProps?	`s3.BucketProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für den Bucket zu überschreiben. Wird ignoriert, wenn ein `existingBucketObj` wird bereitgestellt.
BucketBerechtigungen?	`string[]`	Optionale Bucket-Berechtigungen, die der Lambda Funktion erteilt werden sollen. Eine oder mehrere der folgenden Aktionen können angegeben werden: `Delete`,`Put`,`Read`,`ReadWrite`,`Write`.

Name	Typ	Beschreibung
ExistenzVPC?	`ec2.IVpc`	Eine optionale vorhandene VPC, in der dieses Pattern bereitgestellt werden soll. Bei der Bereitstellung in einer VPC verwendet die Lambda Funktion ENIs in der VPC für den Zugriff auf Netzwerkressourcen, und ein Interface Endpoint wird in der VPC für Amazon SQS erstellt. Wenn eine vorhandene VPC bereitgestellt wird, wird die`deployVpc` kann nicht`true`. Diese Verwendung von`ec2.IVpc`, damit Clients VPCs bereitstellen können, die außerhalb des Stacks vorhanden sind, mithilfe des`ec2.Vpc.fromLookup()` -Methode.

Name	Typ	Beschreibung
DeployVPC?	`boolean`	Erstellen einer neuen VPC basierend auf`vpcProps`, in das dieses Muster bereitgestellt werden soll. Setzen Sie diese Einstellung auf`true`stellt die minimale, privatste VPC bereit, um das Pattern auszuführen: • Ein isoliertes Subnetz in jeder Availability Zone, die vom CDK-Programm verwendet wird. • `enableDnsHostnames` und`enableDnsSupport` werden beide auf`true`. Wenn diese Eigenschaft`true`, dann`existingVpc` kann nicht angegeben werden. Standardeinstellung: `false`.

Name	Typ	Beschreibung
VPCProps?	`ec2.VpcProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die neue VPC zu überschreiben.`enableDnsHostnames` ,`enableDnsSupport` ,`natGateways` und`subnetConfiguration` werden durch das Muster festgelegt, so dass alle Werte für die hier angegebenen Eigenschaften überschrieben werden. Wenn`deployVpc` ist nicht`true`wird diese Eigenschaft ignoriert.
BucketEnvironmentVariableName?	`string`	Optionaler Name für die S3-Bucket-Umgebungsvariable, die für die Lambda Funktion festgelegt wurde.

Muster-Eigenschaften

Name	Typ	Beschreibung
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
S3-Bucket?	`s3.Bucket`	Gibt eine Instanz des S3-Buckets zurück, der durch das Muster erstellt wurde.

Name	Typ	Beschreibung
s3loggingBucket?	s3.Bucket	Gibt eine Instanz des Protokollierungsbuckets zurück, der durch das Muster für den S3-Bucket erstellt wurde.
vpc?	ec2.IVpc	Gibt eine Instanz der VPC zurück, die vom Muster verwendet wird (falls vorhanden). Dies kann eine VPC sein, die durch das Muster oder die VPC erstellt wurde, die dem Musterkonstruktor zur Verfügung gestellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

AWS Lambda-Funktion

- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- Aktivieren der X-Ray blaufverfolgung
- Festlegen von Umgebungsvariablen:
  - `S3_BUCKET_NAME` (Standard)
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Amazon S3 Bucket

- Konfigurieren Sie die Zugriffsprotokollierung für S3 Bucket.

- Aktivieren Sie die serverseitige Verschlüsselung für S3 Bucket mit AWS verwaltetem KMS-Schlüssel.
- Aktivieren Sie die Versionierung für S3 Bucket.
- Erlauben Sie keinen öffentlichen Zugriff für S3 Bucket.
- Behalten Sie den S3-Bucket beim Löschen des CloudFormation-Stacks bei.
- Erzwingen der Verschlüsselung von Daten während der Übertragung
- Wendet die Lebenszyklusregel an, um nicht aktuelle Objektversionen nach 90 Tagen in den Glacier-Speicher zu verschieben.

Architecture

GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-konstrukte/aws-lambda-s3

aws-lambda-ssmstringparameter

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische

VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_lambda_ssm_string_parameter`
Typoskript	`@aws-solutions-constructs/aws-lambda-ssmstringparameter`
Java	`software.amazon.awsconstructs.services.lambdassmstringparameter`

Overview

Dieses AWS Solutions Construct implementiert die AWS Lambda Funktion und den Parameter AWS Systems Manager Parameter Store String mit den geringsten Berechtigungen.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`LambdaToSsmstringparameterProps`

Muster-Konstrukt

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird bereitgestellt.
ExistingStringParameterObj?	`ssm.StringParameter`	Vorhandene Instanz des SSM String-Parameterobjekts, die sowohl diese als

Name	Typ	Beschreibung
		auch`stringParameterProps` führt zu einem -Fehler.
StringParameterProps?	`ssm.StringParameterProps`	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für den SSM-String-Parameter zu überschreiben. Wenn`existingStringParameterObj` nicht gesetzt ist,`stringParameterProps` ist erforderlich. Die einzige unterstützte`ssm.StringParameterProps.type` ist`STRING`, wenn ein anderer Wert angegeben wird, wird er überschrieben.
StringParameterEnvironmentVariableName?	`string`	Optionaler Name für die Umgebungsvariable SSM String-Parameter, die für die Lambda Funktion festgelegt wurde.

Name	Typ	Beschreibung
ExistenzVPC?	`ec2.IVpc`	Eine optionale vorhandene VPC, in der dieses Pattern bereitgestellt werden soll. Bei der Bereitstellung in einer VPC verwendet die Lambda Funktion ENIs in der VPC, um auf Netzwerkressourcen zuzugreifen, und im Parameter VPC for AWS Systems Manager wird ein Interface Endpoint erstellt. Wenn eine vorhandene VPC bereitgestellt wird, wird die `deployVpc` kann nicht `true`. Diese Verwendung von `ec2.IVpc`, damit Clients VPCs bereitstellen können, die außerhalb des Stacks vorhanden sind, mithilfe des `ec2.Vpc.fromLookup()` -Methode.

Name	Typ	Beschreibung
VPCProps?	`ec2.VpcProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die neue VPC zu überschreiben.`enableDnsHostnames` ,`enableDnsSupport` ,`natGateways` und`subnetConfiguration` werden durch das Muster festgelegt, so dass alle Werte für die hier angegebenen Eigenschaften überschrieben werden. Wenn`deployVpc` ist nicht`true`wird diese Eigenschaft ignoriert.

Name	Typ	Beschreibung
DeployVPC?	`boolean`	Ob eine neue VPC basierend auf`vpcProps`, in das dieses Muster bereitgestellt werden soll. Setzen Sie diese Einstellung auf`true`stellt die minimale, privatste VPC bereit, um das Pattern auszuführen: • Ein isoliertes Subnetz in jeder Availability Zone, die vom CDK-Programm verwendet wird. • `enableDnsHostnames` und`enableDnsSupport` werden beide auf`true`. Wenn diese Eigenschaft auf`true`, dann`existingVpc` kann nicht angegeben werden. Standardeinstellung: `false`.
StringParameterPermissions?	`string`	Optionale SSM-String-Parameterberechtigungen, die der Lambda Funktion erteilt werden sollen. Eine der folgenden Angaben kann angegeben werden:`Read`,`ReadWrite` .

Muster-Eigenschaften

Name	Typ	Beschreibung
LambdaFunction	`lambda.Function`	Gibt eine Instanz von`lambda.Function` , die durch das Gebäudemodul erstellt wurde.
StringParameter	`ssm.StringParameter`	Gibt eine Instanz von`ssm.StringParameter` , die durch das Gebäudemodul erstellt wurde.
vpc?	`ec2.IVpc`	Gibt eine Schnittstelle auf der VPC zurück, die vom Muster verwendet wird (falls vorhanden). Dies kann eine VPC sein, die durch das Muster oder die VPC erstellt wurde, die dem Musterkonstruktor zur Verfügung gestellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

AWS Lambda-Funktion

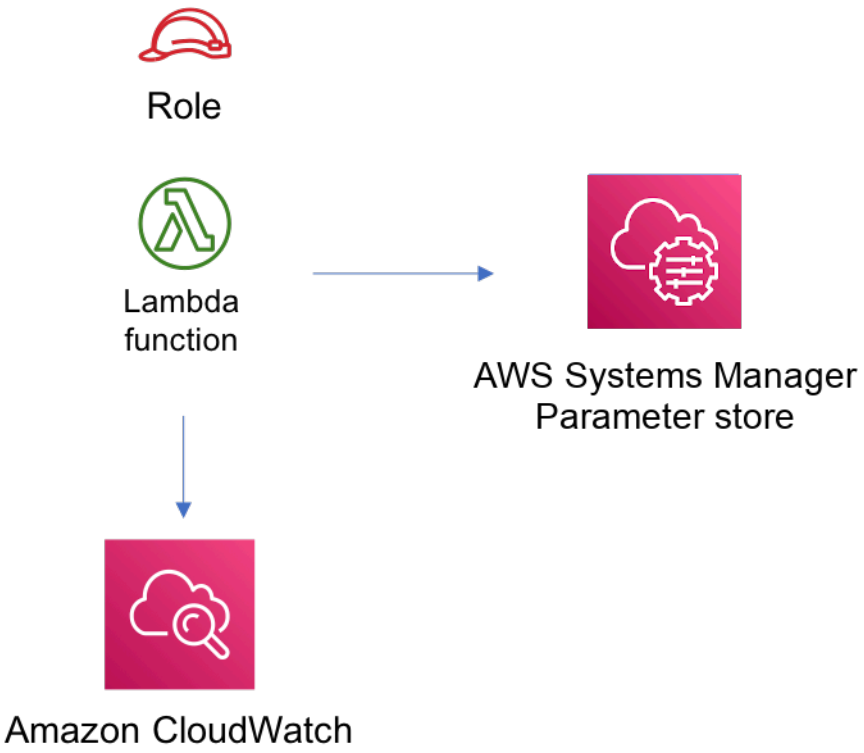
- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- Aktivieren Sie die X-Ray Nachverfolgung.
- Festlegen von Umgebungsvariablen:

- `SSM_STRING_PARAMETER_NAME` (Standard)
- `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Zeichenfolge des Amazon AWS Systems Manager Parameterspeichers

- Aktivieren Sie den schreibgeschützten Zugriff für die zugehörige AWS Lambda Funktion.
- Erstellt einen neuen SSM String-Parameter mit den angegebenen Werten.
- Behalten Sie den SSM-String-Parameter beim Löschen des CloudFormation-Stacks bei.

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-lambda-ssm stringparameter

aws-lambda-sagemakerendpunkt

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht denSemantic VersioningModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_lambda_sagemakerendpoint`
Typoskript	`@aws-solutions-constructs/aws-lambda-sagemakerendpoint`
Java	`software.amazon.awsconstructs.services.lambdasagemakerendpoint`

Overview

Dieses AWS Solutions Construct implementiert eine AWS Lambda Funktion, die mit einem Amazon Sagemaker Endpoint verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`LambdaToSagemakerEndpointProps`

Muster Konstrukt Requisiten

Name	Typ	Beschreibung
Gibt es LambdaObj?	lambda.Function	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch lambdaFunctionProps führt zu einem -Fehler.
LambdaFunctionProps?	lambda.FunctionProps	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben.
ExistingSageMakerEndpointObj?	sagemaker.CfnEndpoint	Ein optionaler, bestehender Sagemaker Enpoint zum Einsatz. Bereitstellung sowohl dieser als auch endpointProps führt zu einem -Fehler.
ModelProps?	sagemaker.CfnModelProps | any	Vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für das Sagemaker-Modell zu überschreiben. Mindestens smodelProps.primaryContainer muss bereitgestellt werden, um ein Modell zu erstellen. Standardmäßig erstellt das Muster eine Rolle mit den mindestens erforderlichen Berechtigungen. Der Client kann jedoch eine benutzerdefinierte Rolle mit zusätzlichen Funktione

Name	Typ	Beschreibung
		n mithilfe von`modelProp s.executionRoleArn` .
EndpointConfigProps?	`sagemaker.CfnEndpo intConfigProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Sagemaker-Endpoint -Konfiguration zu überschre iben.
EndpointProps?	`sagemaker.CfnEndpo intProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für den Sagemaker-Endpunkt zu überschreiben.
ExistenzVPC?	`ec2.IVpc`	Eine optionale vorhandene VPC, in der dieses Konstrukt bereitgestellt werden soll. Bei der Bereitstellung in einer VPC verwenden die Lambda Funktion und Sagemaker Endpoint ENIs in der VPC, um auf Netzwerkressourcen zuzugreifen. Ein Interface Endpoint wird in der VPC für Amazon Sagemaker Runtime und Amazon S3 VPC Endpoint erstellt. Wenn eine vorhanden e VPC bereitgestellt wird, wird die`deployVpc` -Eigenschaft kann nicht`true`.

Name	Typ	Beschreibung
VPCProps?	ec2.VpcProps	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die neue VPC zu überschreiben.enableDnsHostnames ,enableDnsSupport ,natGateways undsubnetConfiguration vom Konstrukt festgelegt werden, so dass alle Werte für die hier angegebenen Eigenschaften überschrieben werden. WenndeployVpc ist nichttruewird diese Eigenschaft ignoriert.

Name	Typ	Beschreibung
DeployVPC?	`boolean`	Gibt an, ob eine neue VPC auf Basis von`vpcProps`, in das dieses Muster bereitgestellt werden soll. Setzen Sie diese Einstellung auf`true`stellt die minimale, privatste VPC bereit, um das Pattern auszuführen: • Ein isoliertes Subnetz in jeder Availability Zone, die vom CDK-Programm verwendet wird. • `enableDnsHostnames` und`enableDnsSupport` werden beide auf`true`. Wenn diese Eigenschaft auf`true`, dann`existingVpc` kann nicht angegeben werden. Standardeinstellung: `false`.
SageMakerEnvironmentVariableName?	`string`	Optionaler Name für die SageMaker Endpunkt-Umgebungsvariable, die für die Lambda Funktion festgelegt wurde.

Muster-Eigenschaften

Name	Typ	Beschreibung
LambdaFunction	lambda.Function	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
SageMakerEndpoint	sagemaker.CfnEndpoint	Gibt eine Instanz des Sagemaker-Endpunkts zurück, der durch das Muster erstellt wurde.
SageMakerEndpointConfig?	sagemaker.CfnEndpointConfig	Gibt eine Instanz der SageMaker EndpointConfig zurück, die durch das Muster erstellt wurde, wenn `existingSagemakerEndpointObj` wird nicht angegeben.
SageMakerModel?	sagemaker.CfnModel	Gibt eine Instanz des Sagemaker-Modells zurück, das durch das Muster erstellt wurde, wenn `existingSagemakerEndpointObj` wird nicht angegeben.
vpc?	`ec2.IVpc`	Gibt eine Instanz der VPC zurück, die durch das Muster erstellt wurde, wenn `deployVpc` ist `true`, oder wenn `existingVpc` wird bereitgestellt.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

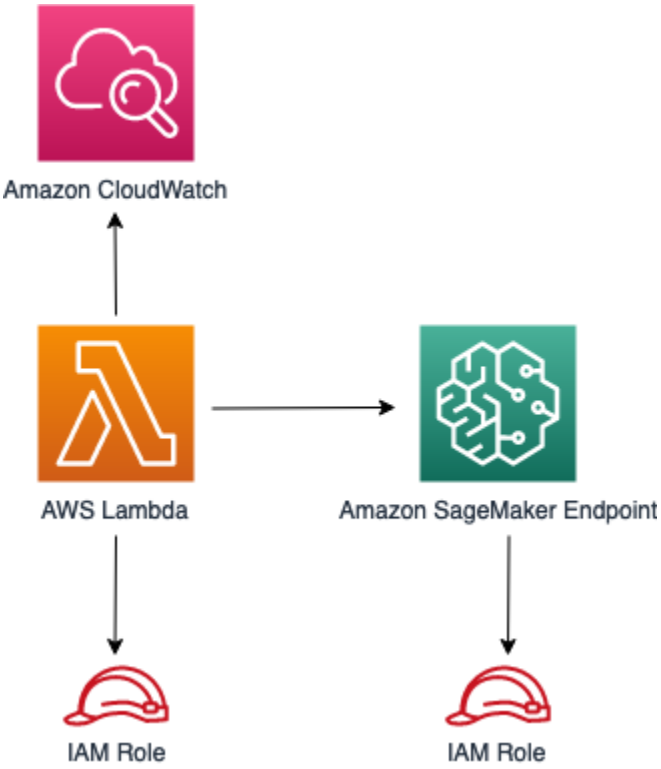
AWS Lambda-Funktion

- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- Erlauben Sie der Funktion, den Sagemaker-Endpunkt für Abschlüsse aufzurufen.
- Konfigurieren Sie die Funktion für den Zugriff auf Ressourcen in der VPC, auf der der Sagemaker-Endpunkt bereitgestellt wird.
- Aktivieren Sie X-Ray Nachverfolgung.
- Festlegen von Umgebungsvariablen:
  - `SAGEMAKER_ENDPOINT_NAME` (Standard)
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Amazon SageMaker-Endpunkt

- Konfigurieren Sie eingeschränkte Berechtigungen zum Erstellen von Sagemaker-Ressourcen.
- Stellen Sie Sagemaker-Modell, EndpointConfig und Endpoint bereit.
- Konfigurieren Sie den Sagemaker-Endpunkt, der in einer VPC bereitgestellt werden soll.
- Stellen Sie die VPC Endpunkt- und Sagemaker-Runtime-VPC-Schnittstelle bereit.

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-lambda-sagemakerendpoint

aws-lambda-Sekretsmanager

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_lambda_secretsmanager`
Typoskript	`@aws-solutions-constructs/aws-lambda-secretsmanager`
Java	`software.amazon.awsconstructs.services.lambdasecretsmanager`

Overview

Dieses AWS Solutions Construct implementiert die AWS Lambda Funktion und das AWS Secrets Manager Geheimnis mit den am wenigsten privilegierten Berechtigungen.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsLambdaToSecretsmanagerProps

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
Gibt es LambdaObj?	lambda.Function	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auchlambdaFunctionProps führt zu einem -Fehler.
LambdaFunctionProps?	lambda.FunctionProps	Der Benutzer hat Requisiten bereitgestellt, um die Standard-Requisiten für die Lambda Funktion zu überschreiben.
SecretProps?	secretsmanager.SecretProps	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisiten für Secrets Manager zu überschreiben.
ExistierenSecretObj?	secretsmanager.Secret	Vorhandene Instanz von Secrets Manager geheimen Objekt, Wenn dies festgeleg

Name	Typ	Beschreibung
		t ist, dann die`secretPro ps` wird ignoriert.
GrantWriteAccess?	`boolean`	Optionaler Schreibzugriff auf das Geheimnis für die Lambda Funktion (standardmäßig schreibgeschützt).
SecretEnvironmentVariableName?	`string`	Optionaler Name für die geheime Umgebungsvariable Secrets Manager, die für die Lambda Funktion festgelegt wurde.
ExistenzVPC?	`ec2.IVpc`	Eine optionale vorhandene VPC, in der dieses Pattern bereitgestellt werden soll. Bei der Bereitstellung in einer VPC verwendet die Lambda Funktion ENIs in der VPC für den Zugriff auf Netzwerkressourcen, und in der VPC für AWS Secrets Manager wird ein Interface Endpoint erstellt. Wenn eine vorhandene VPC bereitgestellt wird, wird die`deployVpc` kann nicht`true`. Diese Verwendung von`ec2.IVpc`, damit Clients VPCs bereitstellen können, die außerhalb des Stacks vorhanden sind, mithilfe des`ec2.Vpc.fromLookup()` -Methode.

Name	Typ	Beschreibung
VPCProps?	ec2.VpcProps	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die neue VPC zu überschreiben.enableDnsHostnames ,enableDnsSupport ,natGateways , undsubnetConfiguration werden durch das Muster festgelegt, so dass alle Werte für die hier angegebenen Eigenschaften überschrieben werden. WenndeployVpc ist nichttruewird diese Eigenschaft ignoriert.

Name	Typ	Beschreibung
DeployVPC?	`boolean`	So erstellen Sie eine neue VPC basierend auf`vpcProps`, in das dieses Muster bereitgestellt werden soll. Setzen Sie diese Einstellung auf`true`stellt die minimale, privatste VPC bereit, um das Pattern auszuführen: • Ein isoliertes Subnetz in jeder Availability Zone, die vom CDK-Programm verwendet wird • `enableDnsHostnames` und`enableDnsSupport` werden beide auf`true` Wenn diese Eigenschaft`true`, dann`existingVpc` kann nicht angegeben werden. Standardeinstellung: `false`.

Muster-Eigenschaften

Name	Typ	Beschreibung
LambdaFUNction	`lambda.Function`	Gibt eine Instanz von`lambda.Function` erstellt durch das Gebäudemodul.
Secret	`secretsmanager.Secret`	Gibt eine Instanz von`secretsmanager.Sec`

Name	Typ	Beschreibung
		`ret` erstellt durch das Gebäudemodul.
vpc?	`ec2.IVpc`	Gibt eine Schnittstelle auf der VPC zurück, die vom Muster verwendet wird (falls vorhanden). Dies kann eine VPC sein, die durch das Muster oder die VPC erstellt wurde, die dem Musterkon struktor zur Verfügung gestellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

AWS Lambda-Funktion

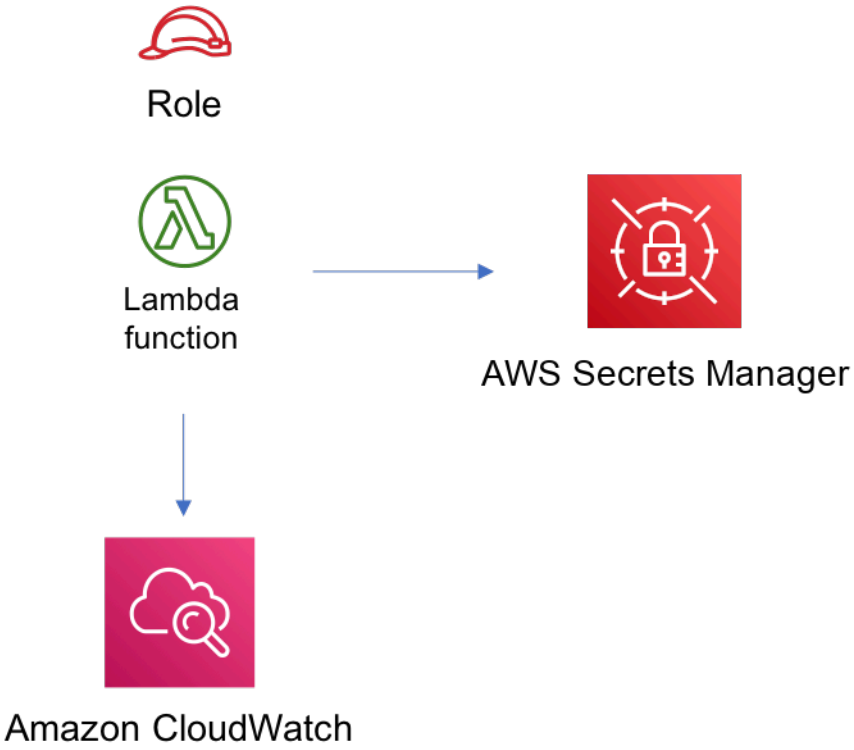
- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- Aktivieren Sie X-Ray Nachverfolgung.
- Festlegen von Umgebungsvariablen:
  - (Standard) SECRET_ARN, die den ARN des Geheimnisses als Rückgabe von CDK enthältSecretArnproperty
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Amazon Secrets Manager-Geheimnis

- Aktivieren des schreibgeschützten Zugriffs für die zugehörige AWS Lambda Funktion
- Aktivieren der serverseitigen Verschlüsselung mithilfe eines standardmäßigen KMS-Schlüssels für das Konto und die Region

- Erstellt ein neues Geheimnis:
  - (Standard) zufälliger Name
  - (Standard) Zufallswert
- Behalten Sie das Geheimnis beim Löschen des CloudFormation -Stacks

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-lambda-secretsmanager

aws-lambda-sns

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische Versionierung-Modell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_lambda_sns`
Typoskript	`@aws-solutions-constructs/aws-lambda-sns`
Java	`software.amazon.awsconstructs.services.lambdasns`

Overview

Dieses AWS Solutions Construct implementiert eine AWS Lambda Funktion, die mit einem Amazon SNS -Thema verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`LambdaToSnsProps`

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird bereitgestellt.
Bestehend TopicObj?	`sns.Topic`	Vorhandene Instanz des SNS Topic-Objekts, die sowohl diese als auch`topicProps` führt zu einem -Fehler.

Name	Typ	Beschreibung
TopicProps?	`sns.TopicProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für das SNS-Thema zu überschreiben.
ExistenzVPC?	`ec2.IVpc`	Eine optionale vorhandene VPC, in der dieses Pattern bereitgestellt werden soll. Bei der Bereitstellung in einer VPC verwendet die Lambda Funktion ENIs in der VPC für den Zugriff auf Netzwerkressourcen, und ein Interface Endpoint wird in der VPC für Amazon SQS erstellt. Wenn eine vorhandene VPC bereitgestellt wird, wird die`deployVpc` kann nicht`true`. Diese Verwendung von`ec2.IVpc`, damit Clients VPCs bereitstellen können, die außerhalb des Stacks vorhanden sind, mithilfe des`ec2.Vpc.fromLookup()` -Methode.

Name	Typ	Beschreibung
DeployVPC?	`boolean`	So erstellen Sie eine neue VPC`vpcProps`, in das dieses Muster bereitgestellt werden soll. Setzen Sie diese Einstellung auf`true`stellt die minimale, privatste VPC bereit, um das Pattern auszuführen: • Ein isoliertes Subnetz in jeder Availability Zone, die vom CDK-Programm verwendet wird. • `enableDnsHostnames` und`enableDnsSupport` werden beide auf`true`. Wenn diese Eigenschaft`true`, dann`existingVpc` kann nicht angegeben werden. Standardeinstellung: `false`.

Name	Typ	Beschreibung
VPCProps?	`ec2.VpcProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die neue VPC zu überschreiben.`enableDnsHostnames` ,`enableDnsSupport` ,`natGateways` und`subnetConfiguration` werden durch das Muster festgelegt, so dass alle Werte für die hier angegebenen Eigenschaften überschrieben werden. Wenn`deployVpc` ist nicht`true`wird diese Eigenschaft ignoriert.
TopicArnEnvironmentVariableName?	`string`	Optionaler Name für die ARN Umgebungsvariable des SNS-Thema, die für die Lambda Funktion festgelegt wurde.
TopicNameEnvironmentVariableName?	`string`	Optionaler Name für die Umgebungsvariable SNS-Themenname, die für die Lambda Funktion festgelegt wurde.

Muster-Eigenschaften

Name	Typ	Beschreibung
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
snsTopic	`sns.Topic`	Gibt eine Instanz des SNS-Themas zurück, das durch das Muster erstellt wurde.
vpc?	`ec2.IVpc`	Gibt eine Instanz der VPC zurück, die vom Muster verwendet wird (falls vorhanden). Dies kann eine VPC sein, die durch das Muster oder die VPC erstellt wurde, die dem Musterkon struktor zur Verfügung gestellt wurde.

Standardeinstellung

Die Out-of-the-Box-Implementierung des Construct ohne Überschreibung setzt die folgenden Standardwerte:

AWS Lambda-Funktion

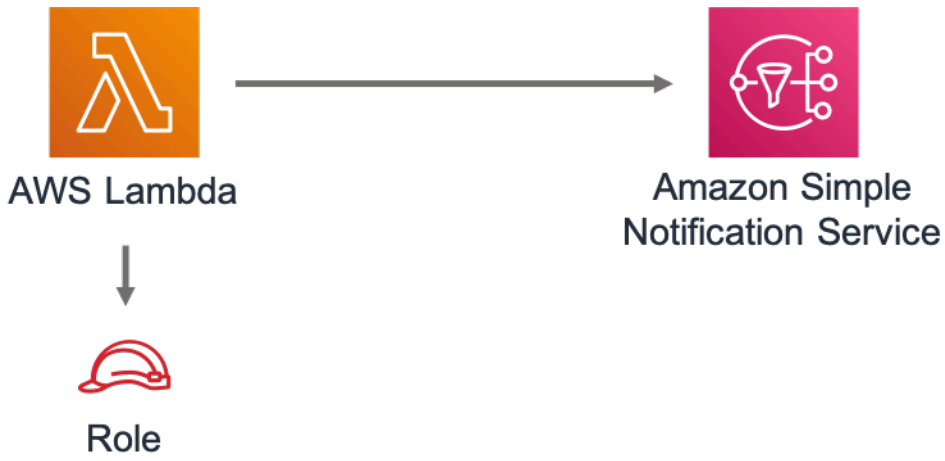
- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- Aktivieren Sie X-Ray Nachverfolgung.
- Festlegen von Umgebungsvariablen:
  - `SNS_TOPIC_NAME` (Standard)

- `SNS_TOPIC_ARN` (Standard)
- `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Amazon SNS-Thema

- Konfigurieren Sie Zugriffsberechtigungen mit den geringsten Berechtigungen für das SNS-Thema.
- Aktivieren Sie die serverseitige Verschlüsselung mit dem verwalteten KMS-Schlüssel von AWS.
- Erzwingen der Verschlüsselung von Daten während der Übertragung

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-konstrukte/aws-lambda-sns

aws-lambda-sqs

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische

VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_lambda_sqs`
Typoskript	`@aws-solutions-constructs/aws-lambda-sqs`
Java	`software.amazon.awsconstructs.services.lambdasqs`

Overview

Dieses AWS Solutions Construct implementiert eine AWS Lambda Funktion, die mit einer Amazon SQS Warteschlange verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsLambdaToSqsProps

Muster-Konstrukt

Name	Typ	Beschreibung
Gibt es LambdaObj?	lambda.Function	Eine optionale, vorhanden e Lambda Funktion, die anstelle der Standardfunktion verwendet wird. Bereitste llung sowohl dieser als auchlambdaFunctionProp s führt zu einem -Fehler.
LambdaFunctionProps?	lambda.FunctionProps	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben.
BestehenQueueObj?	sqs.Queue	Eine optionale, vorhanden e SQS-Warteschlange, die anstelle der Standardw arteschlange verwendet werden soll. Bereitste llung sowohl dieser als

Name	Typ	Beschreibung
		auchqueueProps führt zu einem -Fehler.
QueueProps?	sqs.QueueProps	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die SQS-Warteschlange zu überschreiben.
EnableQueueSurging?	boolean	Gibt an, ob der Lambda Funktion zusätzliche Berechtigungen erteilt werden sollen, um die SQS-Warteschlange zu bereinigen. Standardeinstellung: false.
DeployDeadletterQueue?	boolean	Ob eine sekundäre Warteschlange erstellt wird, die als Warteschlange für unzustellbare Nachrichten verwendet wird. Standardeinstellung: true.
DeadLetterQueueProps?	sqs.QueueProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardrequisiten für die Warteschlange für tote Buchstaben. Wird nur verwendet, wenndeployDeadLetterQueue wird auf „true“ festgelegt.

Name	Typ	Beschreibung
MaxReceiveCount?	`number`	Die Anzahl, wie oft eine Nachricht nicht in die Warteschlange entfernt werden kann, bevor sie in die Warteschlange für unzustell bare Nachrichten verschoben wird. Standardeinstellung: 15.
ExistenzVPC?	`ec2.IVpc`	Eine optionale vorhandene VPC, in der dieses Pattern bereitgestellt werden soll. Bei der Bereitstellung in einer VPC verwendet die Lambda Funktion ENIs in der VPC, um auf Netzwerkr essourcen zuzugreifen, und in der VPC für Amazon SQS wird ein Interface Endpoint erstellt. Wenn eine vorhanden e VPC bereitgestellt wird, wird die`deployVpc` kann nicht`true`. Ein`ec2.IVpc`wird verwendet, um Clients die Bereitstellung von VPCs zu ermöglichen, die außerhalb des Stacks vorhanden sind, mithilfe der`ec2.Vpc.f romLookup()` -Methode.

Name	Typ	Beschreibung
DeployVPC?	`boolean`	Erstellen Sie, ob eine neue VPC basierend auf`vpcProps`, in das dieses Muster bereitgestellt werden soll. Setzen Sie diese Einstellung auf`true`stellt die minimale, privatste VPC bereit, um das Pattern auszuführen: • Ein isoliertes Subnetz in jeder Availability Zone, die vom CDK-Programm verwendet wird • `enableDnsHostnames` und`enableDnsSupport` werden beide auf`true` Wenn diese Eigenschaft`true`, dann`existingVpc` kann nicht angegeben werden. Standardeinstellung: `false`.

Name	Typ	Beschreibung
VPCProps?	`ec2.VpcProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die neue VPC zu überschreiben.`enableDnsHostnames` ,`enableDnsSupport` ,`natGateways` , und`subnetConfiguration` durch das Muster festgelegt werden, so dass alle Werte für die hier angegebenen Eigenschaften überschrieben werden. Wenn`deployVpc` ist nicht`true`wird diese Eigenschaft ignoriert.
WarteschlangenEnvironmentVariablenname?	`string`	Optionaler Name für die SQS-Warteschlangen-URL-Umgebungsvariable, die für die Lambda Funktion festgelegt wurde.

Muster-Eigenschaften

Name	Typ	Beschreibung
DeadLetterQueue?	`sqs.Queue`	Gibt eine Instanz der Warteschlange für toten Buchstaben zurück, die durch das Muster erstellt wurde, wenn eine bereitgestellt wird.

Name	Typ	Beschreibung
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
SQSQueue	`sqs.Queue`	Gibt eine Instanz der SQS-Warteschlange zurück, die durch das Muster erstellt wurde.
vpc?	`ec2.IVpc`	Gibt eine Instanz der VPC zurück, die vom Muster erstellt oder verwendet wird (falls vorhanden). Dies kann eine VPC sein, die durch das Muster erstellt wurde, oder eine VPC, die dem Musterkon struktor zur Verfügung gestellt wird.

Standardeinstellung

Die Out-of-the-Box-Implementierung des Construct ohne Überschreibung setzt die folgenden Standardwerte:

AWS Lambda-Funktion

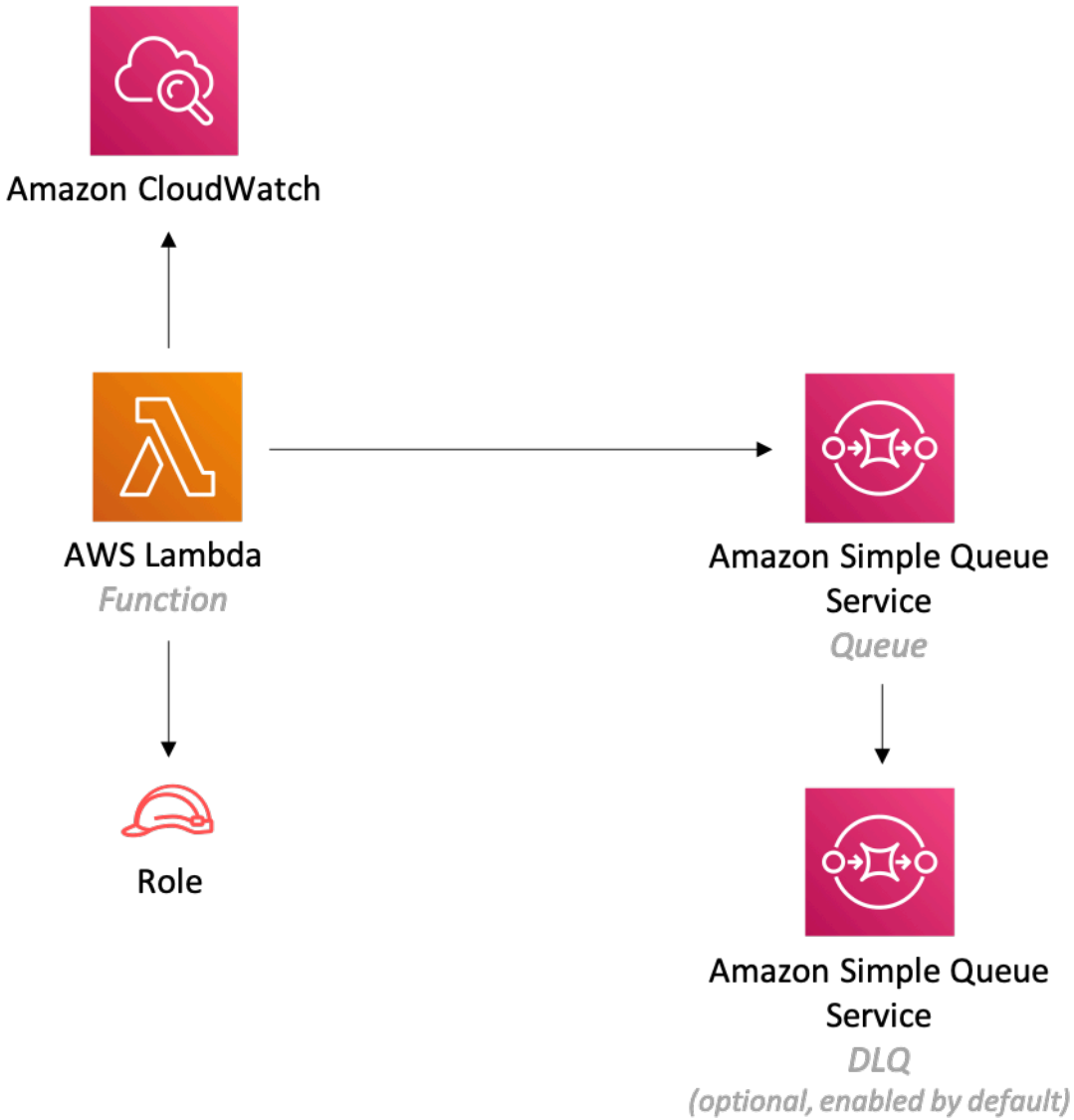
- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- Erlauben Sie der Funktion, nur Nachrichten an die Warteschlange zu senden (das Löschen kann mit de`enableQueuePurge`-Eigenschaft).
- Aktivieren der X-Ray blaufverfolgung
- Festlegen von Umgebungsvariablen:
  - `SQS_QUEUE_URL`

- `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Amazon SQS-Warteschlange

- Bereitstellen der SQS-Warteschlange für unzustellbare Nachrichten für die SQS-Quellwarteschlange
- Aktivieren der serverseitigen Verschlüsselung für SQS-Quellwarteschlange mithilfe von AWS Managed KMS-Schlüssel
- Erzwingen der Verschlüsselung von Daten während der Übertragung

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-lambda-sqs

aws-lambda-sqs-lambda

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht denSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_lambda_sqs_lambda`
Typographische Schrift	`@aws-solutions-constructs/aws-lambda-sqs-lambda`
Java	`software.amazon.awsconstructs.services.lambdasqslambda`

Overview

Dieses AWS Solutions Constructs-Muster implementiert (1) eine AWS Lambda Funktion, die zum Senden von Nachrichten an eine Warteschlange konfiguriert ist, (2) eine Amazon SQS Warteschlange und (3) eine AWS Lambda-Funktion, die für die Verwendung von Nachrichten aus der Warteschlange konfiguriert ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`LambdaToSqsToLambdaProps`

Muster-Konstrukt

Name	Typ	Beschreibung
ExistingProducerLambdaObj?	lambda.Function	Eine optionale, vorhandene Lambda Funktion, die anstelle der Standardfunktion zum Senden von Nachrichten an die Warteschlange verwendet wird. Bereitstellung sowohl dieser als auchproducerLambdaFunctionProps Ein Fehler führt zu einem -Fehler.
ProducerLambdaFunctionProps?	lambda.FunctionProps	Optionale, vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion des Herstellers zu überschreiben.
BestehenQueueObj?	sqs.Queue	Eine optionale, vorhanden e SQS-Warteschlange, die anstelle der Standardw arteschlange verwendet werden soll. Bereitste llung sowohl dieser als auchqueueProps Ein Fehler führt zu einem -Fehler.
QueueProps?	sqs.QueueProps	Optionale vom Benutzer bereitgestellte Eigenscha ften, um die Standarde igenschaften für die SQS-Warteschlange zu überschre iben. Bereitstellung sowohl dieser als auchexistingQ

Name	Typ	Beschreibung
		`ueueObj` Ein Fehler führt zu einem -Fehler.
DeployDeadletterQueue?	`boolean`	Erstellen Sie eine sekundäre Warteschlange, die als Warteschlange für unzustell bare Nachrichten verwendet wird. Standardeinstellung: `true`.
DeadLetterQueueProps?	`sqs.QueueProps`	Optionale, vom Benutzer bereitgestellte Requisite n zum Überschreiben der Standardrequisiten für die Warteschlange für tote Buchstaben. Wird nur verwendet, wenn`deployDea dLetterQueue` -Eigensch aft ist auf`true`.
MaxReceiveCount?	`number`	Die Anzahl, wie oft eine Nachricht nicht erfolgreich entfernt werden kann, bevor sie in die Warteschlange für unzustellbare Nachrichten verschoben wird. Standarde instellung: 15.

Name	Typ	Beschreibung
ExistingConsumerLambdaObj?	`lambda.Function`	Eine optionale, vorhandene Lambda Funktion, die anstelle der Standardfunktion zum Empfangen von Nachrichten aus der Warteschlange verwendet wird. Bereitstellung sowohl dieser als auch `consumerLambdaFunctionProps` Ein Fehler führt zu einem -Fehler.
ConsumerLambdaFunctionProps?	`lambda.FunctionProps`	Optionale, vom Benutzer bereitgestellte Eigenschaften zum Überschreiben der Standardeigenschaften für die Lambda Funktion des Consumer.
WarteschlangenEnvironmentVariablenname?	`string`	Optionaler Name für die SQS-Warteschlangen-URL-Umgebungsvariable, die für die Lambda Funktion des Herstellers festgelegt wurde.

Muster-Eigenschaften

Name	Typ	Beschreibung
VerbraucherLambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion des Verbrauchers zurück, die durch das Muster erstellt wurde.

Name	Typ	Beschreibung
DeadLetterQueue?	sqs.Queue	Gibt eine Instanz der Warteschlange für toten Buchstaben zurück, die durch das Muster erstellt wurde, wenn eine bereitgestellt wird.
ProducerLambdaFunction	lambda.Function	Gibt eine Instanz der Lambda -Funktion des Herstellers zurück, die durch das Muster erstellt wurde.
SQSQueue	sqs.Queue	Gibt eine Instanz der SQS-Warteschlange zurück, die durch das Muster erstellt wurde.

Standardeinstellungen

Die Out-of-the-Box-Implementierung dieses Konstrukts (ohne überschriebene Eigenschaften) erfüllt die folgenden Standardwerte:

Funktionen von AWS Lambda

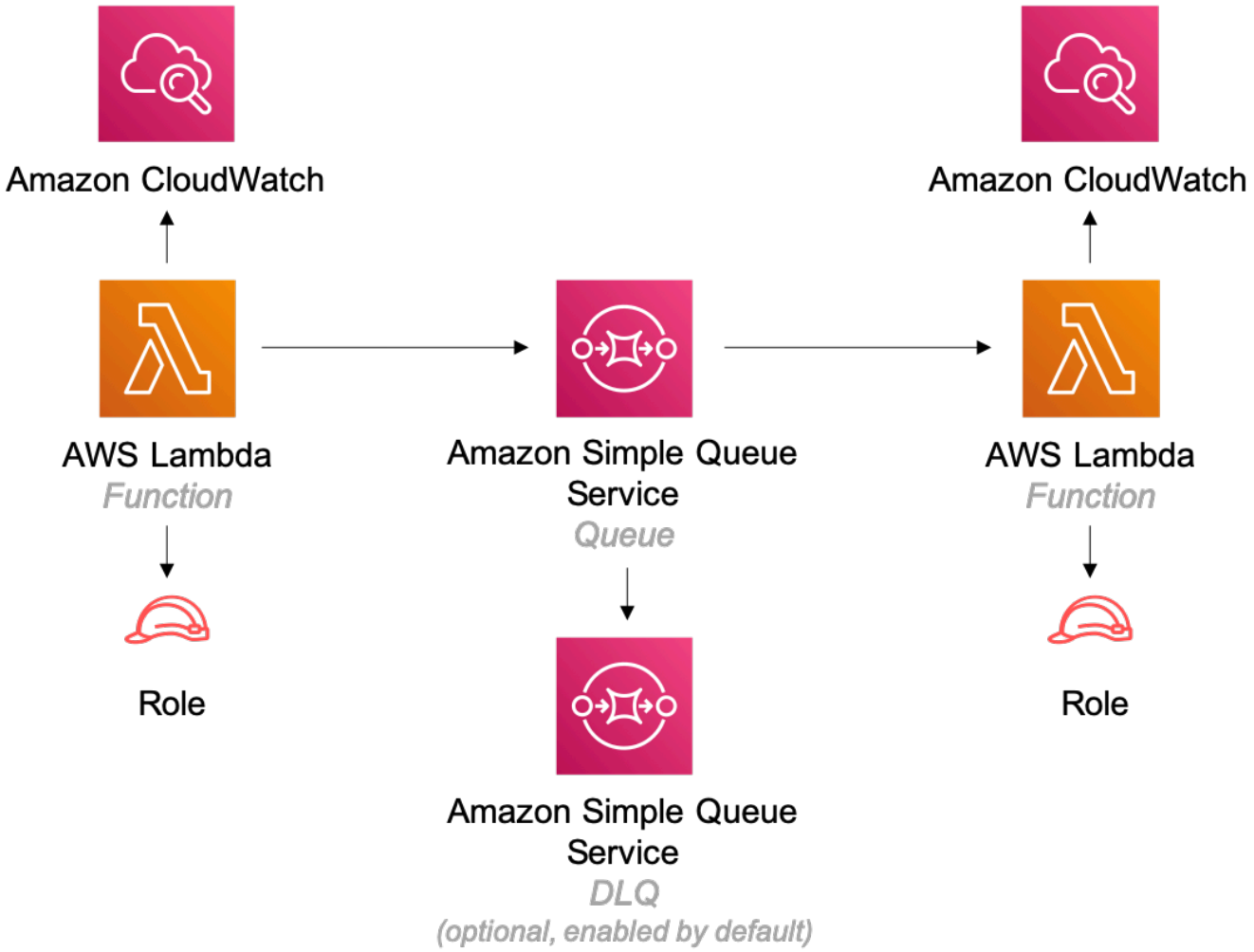
- Konfigurieren Sie die IAM-Rolle für eingeschränkten Zugriffsberechtigungen für Lambda Funktionen.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktionen.
- Aktivieren von X-Ray blaufverfolgung
- Festlegen von Umgebungsvariablen:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Amazon SQS-Warteschlange

- Bereitstellen Sie eine Warteschlange für unzustellbare Nachrichten für die primäre

- Aktivieren Sie die serverseitige Verschlüsselung für die primäre Warteschlange mithilfe eines AWS Managed KMS-Schlüssels.
- Erzwingen der Verschlüsselung von Daten während der Übertragung

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-lambda-sqs-lambda

aws-lambda-Schritt-Funktion

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_lambda_step_function`
Typoskript	`@aws-solutions-constructs/aws-lambda-step-function`
Java	`software.amazon.awsconstructs.services.lambdastepfunction`

Overview

Dieses AWS Solutions Construct implementiert eine AWS Lambda Funktion, die mit einer AWS Step Function verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`LambdaToStepFunctionProps`

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` führt zu einem -Fehler.
LambdaFactionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften,

Name	Typ	Beschreibung
		um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird bereitgestellt.
StateMachineProps	`sfn.StateMachineProps`	Der Benutzer hat Requisiten für die SFN.StateMachine bereitgestellt.
CreateCloudWatchalarms	`boolean`	Gibt an, ob empfohlene CloudWatch Alarme erstellt werden sollen.
LogGroupProps?	`logs.LogGroupProps`	Optionale, vom Benutzer bereitgestellte Requisiten, um die Standardprops für die CloudWatch Logs s-Protokollgruppe zu überschreiben.
StateMachineEnvironmentVariableName	`string`	Optionaler Name für die Systemumgebungsvariable „Step Functions“, die für die Lambda Funktion des Herstellers festgelegt wurde.

Muster-Eigenschaften

Name	Typ	Beschreibung
CloudwatChalarme?	`cloudwatch.Alarm[]`	Gibt eine Liste von einem oder mehreren CloudWatch Alarmen zurück, die durch das Muster erstellt wurden.

Name	Typ	Beschreibung
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
StateMachine	`sfn.StateMachine`	Gibt eine Instanz der Zustandsmaschine zurück, die durch das Muster erstellt wurde.
StateMachineLogGroup	`logs.LogGroup`	Gibt eine Instanz der Protokoll gruppe zurück, die durch das Muster für den Zustandsm aschine erstellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

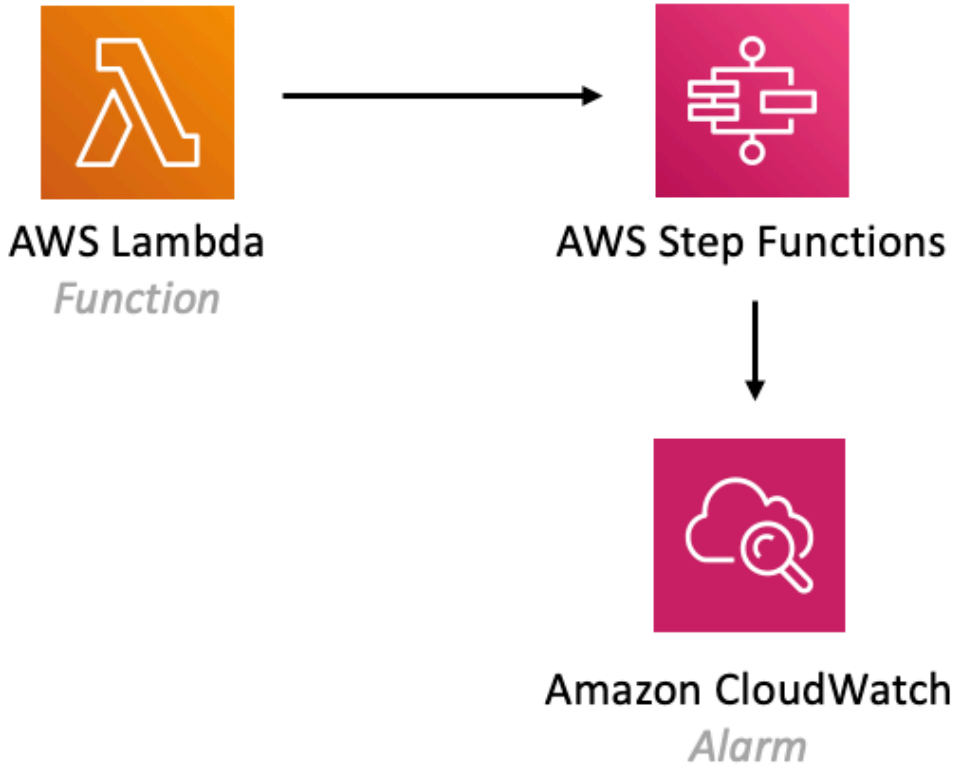
AWS Lambda-Funktion

- Konfigurieren Sie eine IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für die Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktionen.
- Aktivieren der X-Ray blaufverfolgung.
- Festlegen von Umgebungsvariablen:
  - `STATE_MACHINE_ARN` (Standard)
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

AWS Step Functions State Machine

- Stellen Sie bewährte CloudWatch Alarme für die AWS Step Functions State Machine bereit.

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-lambda-step-Funktion

aws-s3-lambda

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_s3_lambda`
Typoskript	`@aws-solutions-constructs/aws-s3-lambda`
Java	`software.amazon.awsconstructs.services.s3lambda`

Overview

Dieses AWS Solutions Construct implementiert einen Amazon S3 Bucket, der mit einer AWS Lambda Funktion verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`S3ToLambdaProps`

Muster-Konstrukt

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird bereitgestellt.
Bestehend BuckeObj?	`s3.Bucket`	Vorhandene Instanz des S3 Bucket-Objekts. Wenn dies vorgesehen ist, dann wird auch die Bereitstellung

Name	Typ	Beschreibung
		von`bucketProps` ist ein Fehler.
BucketProps?	`s3.BucketProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für den Bucket zu überschreiben. Wird ignoriert, wenn ein`existingBucketObj` wird bereitgestellt.
S3EventSourceProps?	`S3EventSourceProps`	Optional vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für S3EventSourceProps

Muster-Eigenschaften

Name	Typ	Beschreibung
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
S3-Bucket?	`s3.Bucket`	Gibt eine Instanz des S3-Buckets zurück, der durch das Muster erstellt wurde.
s3loggingBucket?	`s3.Bucket`	Gibt eine Instanz des Protokollierungsbuckets zurück, der durch das Muster für den S3-Bucket erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

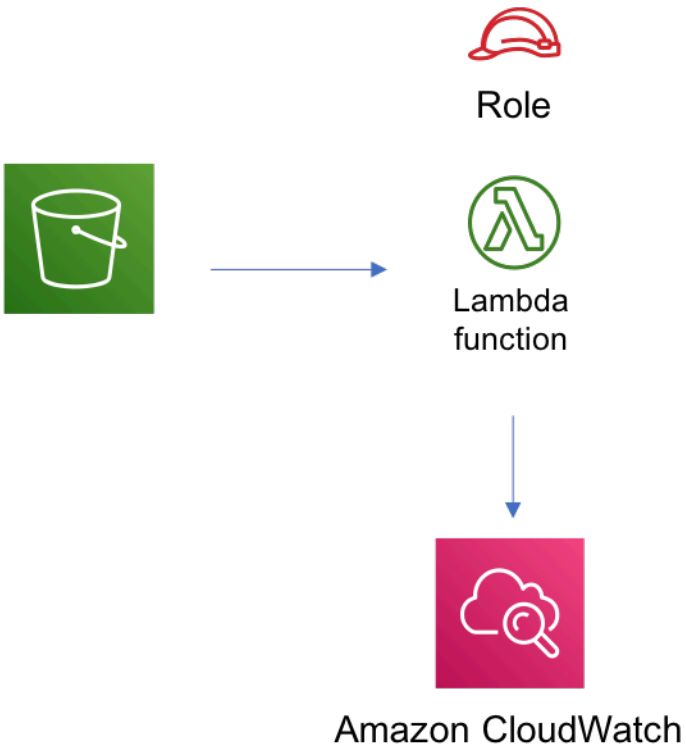
Amazon S3 Bucket

- Konfigurieren Sie die Zugriffsprotokollierung für S3 Bucket.
- Aktivieren Sie die serverseitige Verschlüsselung für S3 Bucket mit AWS verwaltetem KMS-Schlüssel.
- Aktivieren Sie die Versionierung für S3 Bucket.
- Erlauben Sie keinen öffentlichen Zugriff für S3 Bucket.
- Behalten Sie den S3-Bucket beim Löschen des CloudFormation-Stacks bei.
- Erzwingen der Verschlüsselung von Daten während der Übertragung
- Wendet die Lebenszyklusregel an, um nicht aktuelle Objektversionen nach 90 Tagen in den Glacier-Speicher zu verschieben.

AWS Lambda-Funktion

- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- Aktivieren der X-Ray Nachverfolgung
- Festlegen von Umgebungsvariablen:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-konstrukte/aws-s3-lambda

aws-s3-sqs

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht denSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_s3_sqs`
TypoScript	`@aws-solutions-constructs/aws-s3-sqs`
Java	`software.amazon.awsconstructs.services.s3sqs`

Overview

Dieses AWS Solutions Construct implementiert einen Amazon S3 Bucket, der so konfiguriert ist, dass Benachrichtigungen an eine Amazon SQS Warteschlange gesendet werden.

Hier ist eine minimale bereitstellbare Musterdefinition in TypeScript:

```
import { S3ToSqs } from "@aws-solutions-constructs/aws-s3-sqs";

new S3ToSqs(stack, 'S3ToSQSPattern', {});
```

Initializer

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

Parameter

- Bereich`Construct`

- `id` `string`
- `props` `S3ToSqsProps`

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
Bestehend BuckeObj?	`s3.Bucket`	Vorhandene Instanz des S3 Bucket-Objekts Wenn dies zur Verfügung gestellt wird, dann wird auch `bucketProps` ist ein Fehler.
BucketProps?	`s3.BucketProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standard-Requisiten für den S3-Bucket.
s3EventTypes?	`s3.EventType[]`	Die S3-Ereignistypen, die die Benachrichtigung auslösen. Standardeinstellung: `s3.EventType.OBJECT_CREATED` .
s3EventFilters?	`s3.NotificationKeyFilter[]`	Die S3-Objektschlüssel filterregeln bestimmen , welche Objekte dieses Ereignis auslösen. Wenn nicht angegeben, werden keine Filterregeln angewendet.
BestehenQueueObj?	`sqs.Queue`	Eine optionale, vorhandene SQS-Warteschlange, die anstelle der Standardwarteschlange verwendet werden soll. Bereitstellung sowohl dieser als

Name	Typ	Beschreibung
		auchqueueProps führt zu einem -Fehler. Wenn die SQS-Warteschlange verschlüsselt ist, muss der für die Verschlüsselung verwendete KMS-Schlüssel ein vom Kunden verwaltetes CMK sein.
QueueProps?	sqs.QueueProps	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die SQS-Warteschlange zu überschreiben. Wird ignoriert, wenn einexistingQueueObj wird bereitgestellt.
DeadLetterQueueProps?	sqs.QueueProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardrequisiten für die Warteschlange für tote Buchstaben. Wird nur verwendet, wenn diedeployDeadLetterQueue -Eigenschaft auf true festgelegt.
DeployDeadletterQueue?	boolean	Erstellen Sie eine sekundäre Warteschlange, die als Warteschlange für unzustellbare Nachrichten dient. Standardeinstellung: true.

Name	Typ	Beschreibung
MaxReceiveCount?	`number`	Die Anzahl, wie oft eine Nachricht nicht erfolgreich entfernt werden kann, bevor sie in die Warteschlange für unzustellbare Nachrichten verschoben wird. Standarde instellung: 15.
EnableEncryptionWithCustomerManagedKey?	`boolean`	Gibt an, ob ein KMS-Schlüssel verwendet werden soll, der entweder von dieser CDK-App verwaltet oder importiert wird. Beim Importieren eines Verschlüsselungsschlüssels muss dieser in der`encryptionKey` -Eigenschaft für dieses Konstrukt.
encryptionKey?	`kms.Key`	Ein optionaler, vorhandener Verschlüsselungsschlüssel, der anstelle des Standard-Verschlüsselungsschlüssels verwendet werden soll.
EncryptionKeyProps?	`kms.KeyProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für den Verschlüsselungsschlüssel zu überschreiben.

Muster-Eigenschaften

Name	Typ	Beschreibung
SQSQueue	`sqs.Queue`	Gibt eine Instanz der SQS-Warteschlange zurück, die durch das Muster erstellt wurde.
DeadLetterQueue?	`sqs.Queue`	Gibt eine Instanz der Warteschlange für toten Buchstaben zurück, die durch das Muster erstellt wurde, wenn eine bereitgestellt wird.
encryptionKey	`kms.IKey`	Gibt eine Instanz des Verschlüsselungsschlüssels zurück, der durch das Muster erstellt wurde.
S3Bucket?	`s3.Bucket`	Gibt eine Instanz des S3-Buckets zurück, der durch das Muster erstellt wurde.
s3loggingBucket?	`s3.Bucket`	Gibt eine Instanz des Protokollierungsbuckets zurück, der durch das Muster für den S3-Bucket erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon S3 Bucket

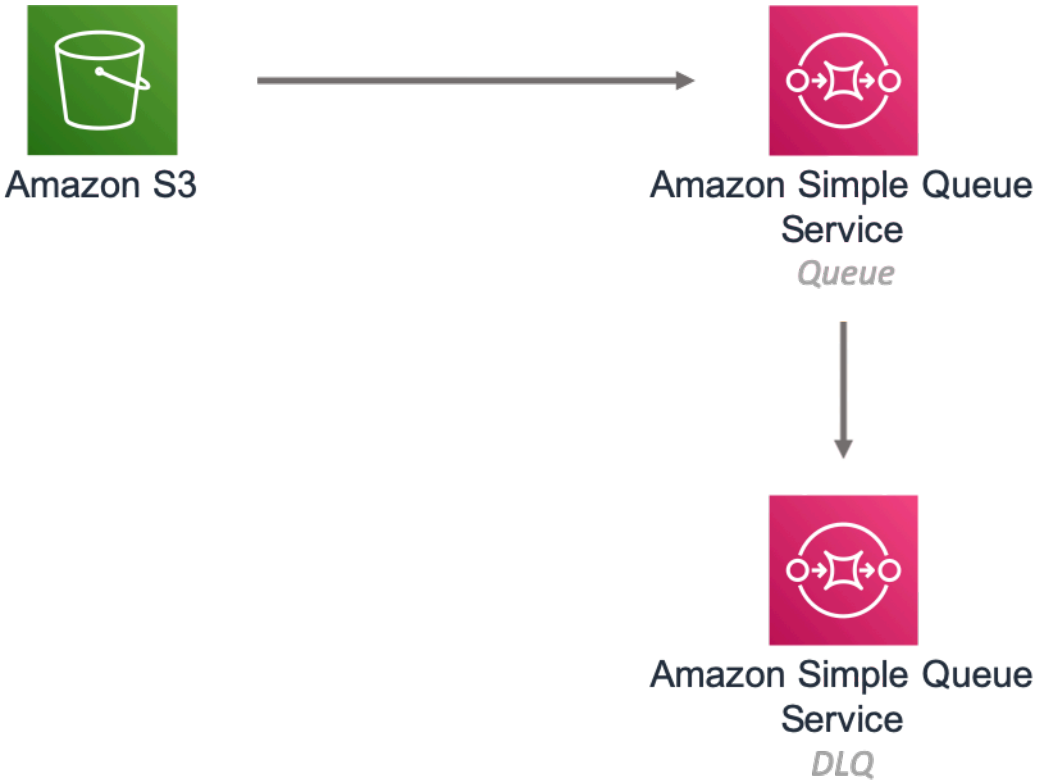
- Konfigurieren der Zugriffsprotokollierung für S3 Bucket
- Aktivieren der serverseitigen Verschlüsselung für S3 Bucket mit AWS verwaltetem KMS-Schlüssel

- Aktivieren der Versionierung für S3 Bucket
- Öffentlichen Zugriff für S3 Bucket nicht zulassen
- Beibehalten des S3-Bucket beim Löschen des CloudFormation -Stacks
- Erzwingen der Verschlüsselung von Daten während der Übertragung
- Wendet die Lebenszyklusregel an, um nicht aktuelle Objektversionen nach 90 Tagen in Glacier-Speicher zu verschieben

Amazon SQS-Warteschlange

- Konfigurieren der Zugriffsberechtigungen für die geringste Berechtigung für SQS-Warteschlange
- Bereitstellen der SQS-Warteschlange für unzustellbare Nachrichten
- Aktivieren der serverseitigen Verschlüsselung für SQS-Warteschlange mithilfe des vom Kunden verwalteten KMS-Schlüssels
- Erzwingen der Verschlüsselung von Daten während der Übertragung

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

	@aws -solutions-constructs/aws-s3-sqs

aws-s3-Schritt-Funktion

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantic VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_s3_step_function`
Typoskript	`@aws-solutions-constructs/aws-s3-step-function`
Java	`software.amazon.awsconstructs.services.s3stepfunction`

Overview

Dieses AWS Solutions Construct implementiert einen Amazon S3 Bucket, der mit einer AWS Step Function verbunden ist.

> **Note**
>
> Dieses Konstrukt verwendet Amazon EventBridge (Amazon CloudWatch Events), um AWS Step Functions auszulösen. EventBridge ist flexibler, aber das Auslösen von Step Functions mit S3-Ereignisbenachrichtigungen hat weniger Latenz und ist kostengünstiger. Wenn Kosten und/oder Latenz ein Problem darstellen, sollten Sie die Bereitstellung`aws-s3-lambda`und`aws-lambda-stepfunctions`anstelle dieses Konstrukt.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`

- `idstring`
- propsS3ToStepFunctionProps

Muster-Konstrukt

Name	Typ	Beschreibung
Bestehend BuckeObj?	s3.IBucket	Vorhandene Instanz des S3 Bucket-Objekts Wenn dies zur Verfügung gestellt wird, dann wird auch`bucketProps` ist ein Fehler.
BucketProps?	s3.BucketProps	Optionale vom Benutzer bereitgestellte Eigenscha ften, um die Standarde igenschaften für den Bucket zu überschreiben. Wird ignoriert, wenn ein`existingB ucketObj` wird bereitgestellt.
StateMachineProps	sfn.StateMachinePr ops	Optional vom Benutzer bereitgestellte Requisiten, um die Standard-Requisite n für SFN.StateMachine zu überschreiben.
EventRuleProps?	events.RuleProps	Optionaler Benutzer hat EventRuleProps bereitges tellt, um die Standardwerte zu überschreiben.
DeployCloudTrail?	`boolean`	Gibt an, ob ein Trail in AWS CloudTrail bereitgestellt wird, um API-Ereignisse in Amazon S3 zu protokollieren. Standardeinstellung: `true`.

Name	Typ	Beschreibung
CreateCloudWatchalarms	`boolean`	Gibt an, ob empfohlene CloudWatch Alarme erstellt werden sollen.
LogGroupProps?	`logs.LogGroupProps`	Optionale, vom Benutzer bereitgestellte Requisiten, um die Standardprops für die CloudWatch Logs s-Protoko llgruppe zu überschreiben.

Muster-Eigenschaften

Name	Typ	Beschreibung
cloudtrail?	`cloudtrail.Trail`	Gibt eine Instanz des durch das Muster erstellten Cloudtrai l zurück.
CloudTrailBucket?	`s3.Bucket`	Gibt eine Instanz des Buckets zurück, der durch das Muster zum Speichern von Cloudtrail Traildaten erstellt wurde.
CloudTrailLoggingBucket?	`s3.Bucket`	Gibt eine Instanz des Protokoll ierungsbuckets zurück, der durch das Muster für den primären Bucket erstellt wurde, der vom Cloudtrail Trail verwendet wird.
Cloudwatchalarms?	`cloudwatch.Alarm[]`	Gibt eine Liste eines oder mehrerer CloudWatch Alarme zurück, die durch das Muster erstellt wurden.

Name	Typ	Beschreibung
S3Eimer?	`s3.Bucket`	Gibt eine Instanz des S3-Buckets zurück, der durch das Muster erstellt wurde.
s3loggingBucket?	`s3.Bucket`	Gibt eine Instanz des Protokollierungsbuckets zurück, der durch das Muster für den S3-Bucket erstellt wurde.
StateMachine	`sfn.StateMachine`	Gibt eine Instanz der Zustandsmaschine zurück, die durch das Muster erstellt wurde.
StateMachineLogGroup	`logs.LogGroup`	Gibt eine Instanz der Protokollgruppe zurück, die durch das Muster für den Zustandsmaschine erstellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon S3 Bucket

- Konfigurieren Sie die Zugriffsprotokollierung für S3 Bucket.
- Aktivieren Sie die serverseitige Verschlüsselung für S3 Bucket mit AWS verwaltetem KMS-Schlüssel.
- Aktivieren Sie die Versionierung für S3 Bucket.
- Erlauben Sie keinen öffentlichen Zugriff für S3 Bucket.
- Behalten Sie den S3-Bucket beim Löschen des CloudFormation-Stacks bei.
- Erzwingen der Verschlüsselung von Daten während der Übertragung
- Wendet die Lebenszyklusregel an, um nicht aktuelle Objektversionen nach 90 Tagen in den Glacier-Speicher zu verschieben.

AWS CloudTrail

- Konfigurieren Sie einen Trail in AWS CloudTrail, um API-Ereignisse in Amazon S3 zu protokollieren, die sich auf den Bucket beziehen, der vom Construct erstellt wurde.

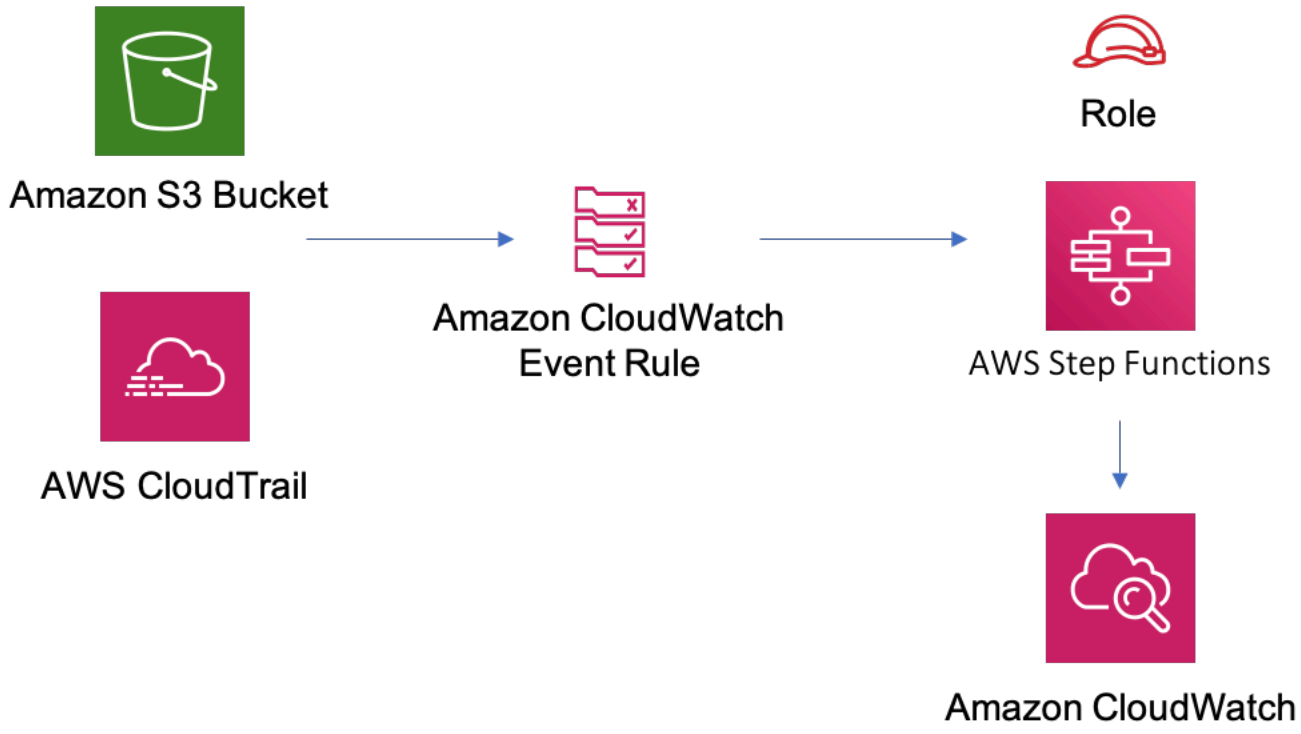
Amazon CloudWatch Events regel

- Erteilen Sie CloudWatch Ereignissen die geringsten Berechtigungen, um die Lambda Funktion auszulösen.

AWS Step Function

- Aktivieren Sie die CloudWatch Protokollierung für API Gateway.
- Stellen Sie Best Practices CloudWatch Alarme für die Step-Funktion bereit.

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:	
	@aws -solutions-Konstrukte/aws-s3-Schritt-Funktion

aws-sns-lambda

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_sns_lambda`
Typographische Schrift	`@aws-solutions-constructs/aws-sns-lambda`
Java	`software.amazon.awsconstructs.services.snslambda`

Overview

Dieses AWS Solutions Construct implementiert ein Amazon SNS, das mit einer AWS Lambda Funktion verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsSnsToLambdaProps

Muster-Konstrukt

Name	Typ	Beschreibung
Gibt es LambdaObj?	lambda.Function	Vorhandene Instanz des Lambda Function-Objekts,

Name	Typ	Beschreibung
		die sowohl diese als auch `lambdaFunctionProps` führt zu einem -Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein `existingLambdaObj` wird bereitgestellt.
Bestehend TopicObj?	`sns.Topic`	Vorhandene Instanz des SNS Topic-Objekts, die sowohl diese als auch `topicProps` führt zu einem -Fehler.
TopicProps?	`sns.TopicProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für das SNS-Thema zu überschreiben.

Muster-Eigenschaften

Name	Typ	Beschreibung
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
snsTopic	`sns.Topic`	Gibt eine Instanz des SNS-Themas zurück, das durch das Muster erstellt wurde.

Standardeinstellungen

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

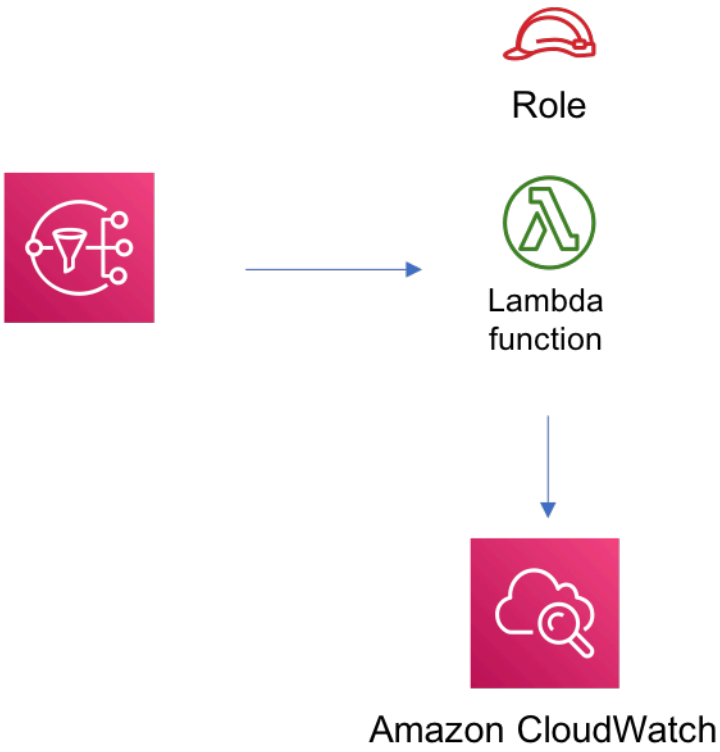
Amazon SNS-Thema

- Konfigurieren Sie Zugriffsberechtigungen mit den geringsten Berechtigungen für das SNS-Thema.
- Aktivieren Sie die serverseitige Verschlüsselung mit dem verwalteten KMS-Schlüssel von AWS.
- Erzwingen der Verschlüsselung von Daten während der Übertragung

AWS Lambda-Funktion

- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.
- Aktivieren Sie die X-Ray blaufverfolgung
- Festlegen von Umgebungsvariablen:
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-constructs/aws-sns-lambda

aws-sns-sqs

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht denSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_sns_sqs`
Typographische Schrift	`@aws-solutions-constructs/aws-sns-sqs`
Java	`software.amazon.awsconstructs.services.snssqs`

Overview

Dieses AWS Solutions Construct implementiert ein Amazon SNS -Thema, das mit einer Amazon SQS Warteschlange verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- BereichConstruct
- idstring
- propsSnsToSqsProps

Muster-Konstrukt Requisiten

Name	Typ	Beschreibung
Bestehend TopicObj?	sns.Topic	Vorhandene Instanz des SNS Topic-Objekts, die sowohl diese als auchtopicProps führt zu einem -Fehler.
TopicProps?	sns.TopicProps	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für das SNS-Thema zu überschreiben. Wird ignoriert, wenn einexistingTopicObj wird bereitgestellt.
BestehenQueueObj?	sqs.Queue	Eine optionale, vorhandene SQS-Warteschlange, die anstelle der Standardwarteschlange verwendet werden soll. Bereitstellung sowohl dieser als auchqueueProps führt zu einem -Fehler.

Name	Typ	Beschreibung
QueueProps?	sqs.QueueProps	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die SQS-Warteschlange zu überschreiben. Wird ignoriert, wenn ein `existingQueueObj` wird bereitgestellt.
DeployDeadletterQueue?	`boolean`	Erstellen einer sekundären Warteschlange, die als Warteschlange für unzustellbare Nachrichten verwendet wird. Standardeinstellung: `true`.
DeadLetterQueueProps?	sqs.QueueProps	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardrequisiten für die Warteschlange für tote Buchstaben. Wird nur verwendet, wenn die `deployDeadLetterQueue` auf true gesetzt.
MaxReceiveCount?	`number`	Die Anzahl, wie oft eine Nachricht nicht erfolgreich entfernt werden kann, bevor sie in die Warteschlange für unzustellbare Nachrichten verschoben wird. Standardeinstellung: 15.

Name	Typ	Beschreibung
EnableEncryptionWithCustomerManagedKey?	`boolean`	Gibt an, ob ein vom Kunden verwalteter Verschlüsselungsschlüssel verwendet werden soll, der entweder von dieser CDK-App verwaltet oder importiert wird. Beim Importieren eines Verschlüsselungsschlüssels muss dieser in der `encryptionKey` -Eigenschaft für dieses Konstrukt.
encryptionKey?	`kms.Key`	Ein optionaler, vorhandener Verschlüsselungsschlüssel, der anstelle des Standard-Verschlüsselungsschlüssels verwendet werden soll.
EncryptionKeyProps?	`kms.KeyProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für den Verschlüsselungsschlüssel zu überschreiben.

Muster-Eigenschaften

Name	Typ	Beschreibung
snsTopic	`sns.Topic`	Gibt eine Instanz des SNS-Themas zurück, das durch das Muster erstellt wurde.
encryptionKey	`kms.Key`	Gibt eine Instanz des Verschlüsselungsschlüssels

Name	Typ	Beschreibung
		zurück, der durch das Muster erstellt wurde.
SQSQueue	sqs.Queue	Gibt eine Instanz der SQS-Warteschlange zurück, die durch das Muster erstellt wurde.
DeadLetterQueue?	sqs.Queue	Gibt eine Instanz der Warteschlange für toten Buchstaben zurück, die durch das Muster erstellt wurde, wenn eine bereitgestellt wird.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

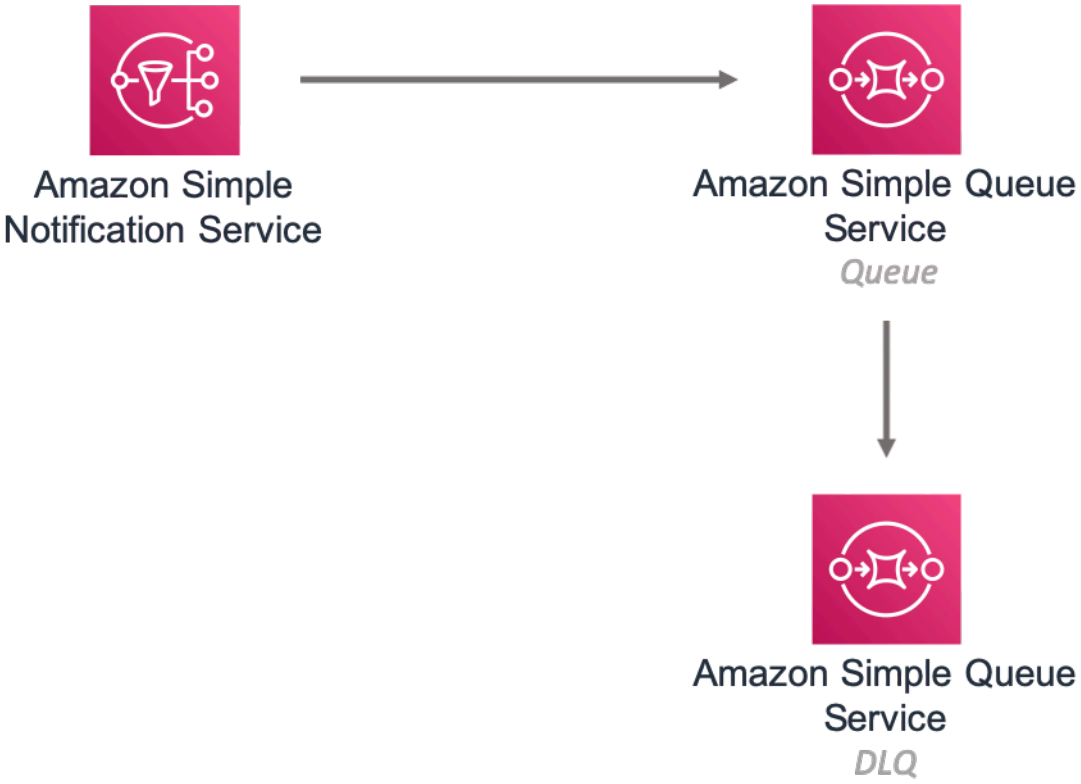
Amazon SNS-Thema

- Konfigurieren Sie Zugriffsberechtigungen mit den geringsten Berechtigungen für das SNS-Thema.
- Aktivieren Sie die serverseitige Verschlüsselung mit dem verwalteten KMS-Schlüssel von AWS.
- Erzwingen der Verschlüsselung von Daten während der Übertragung

Amazon SQS-Warteschlange

- Konfigurieren Sie Zugriffsberechtigungen für die niedrigste Berechtigung für SQS-Warteschlange.
- Bereitstellen der Warteschlange für unzustellbare Nachrichten für die SQS-Quellwarteschlange
- Aktivieren der serverseitigen Verschlüsselung für SQS-Warteschlange mithilfe des vom Kunden verwalteten KMS-Schlüssels
- Erzwingen der Verschlüsselung von Daten während der Übertragung

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-constructs/aws-sns-sqs

aws-sqs-lambda

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht denSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie ein Upgrade auf eine neuere Version dieses Pakets durchführen.

Hinweis: Um eine ordnungsgemäße Funktionalität sicherzustellen, müssen die AWS Solutions Constructs Pakete und AWS CDK-Pakete in Ihrem Projekt dieselbe Version aufweisen.

Sprache	Paket
Python	`aws_solutions_constructs.aws_sqs_lambda`
Typographische Schrift	`@aws-solutions-constructs/aws-sqs-lambda`
Java	`software.amazon.awsconstructs.services.sqslambda`

Overview

Dieses AWS Solutions Construct implementiert eine Amazon SQS Warteschlange, die mit einer AWS Lambda Funktion verbunden ist.

Hier ist eine minimale bereitstellbare Musterdefinition 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);
```

Parameter

- Bereich`Construct`
- id`string`
- props`SqsToLambdaProps`

Muster-Requisiten

Name	Typ	Beschreibung
Gibt es LambdaObj?	`lambda.Function`	Vorhandene Instanz des Lambda Function-Objekts, die sowohl diese als auch`lambdaFunctionProps` verursacht einen Fehler.
LambdaFunctionProps?	`lambda.FunctionProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die Lambda Funktion zu überschreiben. Wird ignoriert, wenn ein`existingLambdaObj` wird bereitgestellt.
BestehenQueueObj?	`sqs.Queue`	Eine optionale, vorhandene SQS-Warteschlange, die anstelle der Standardwarteschlange verwendet werden soll. Bereitstellung sowohl dieser als auch`queueProps` verursacht einen Fehler.

Name	Typ	Beschreibung
QueueProps?	`sqs.QueueProps`	Optionale vom Benutzer bereitgestellte Eigenschaften, um die Standardeigenschaften für die SQS-Warteschlange zu überschreiben. Wird ignoriert, wenn ein `existingQueueObj` wird bereitgestellt.
DeployDeadletterQueue?	`boolean`	Erstellen Sie eine sekundäre Warteschlange, die als Warteschlange für unzustellbare Nachrichten dient. Standardeinstellung: `true`.
DeadLetterQueueProps?	`sqs.QueueProps`	Optionale, vom Benutzer bereitgestellte Requisiten zum Überschreiben der Standardrequisiten für die Warteschlange für tote Buchstaben. Wird nur verwendet, wenn die `deployDeadLetterQueue` -Eigenschaft auf „true“ festgelegt ist.
MaxReceiveCount?	`number`	Die Anzahl, wie oft eine Nachricht nicht erfolgreich entfernt werden kann, bevor sie in die Warteschlange für unzustellbare Nachrichten verschoben wird. Standardeinstellung: 15.

Muster-Eigenschaften

Name	Typ	Beschreibung
DeadLetterQueue?	`sqs.Queue`	Gibt eine Instanz der Warteschlange für toten Buchstaben zurück, die durch das Muster erstellt wurde, wenn eine bereitgestellt wird.
LambdaFunction	`lambda.Function`	Gibt eine Instanz der Lambda Funktion zurück, die durch das Muster erstellt wurde.
SQSQueue	`sqs.Queue`	Gibt eine Instanz der SQS-Warteschlange zurück, die durch das Muster erstellt wurde.

Standardeinstellung

Die vorgefertigte Implementierung dieses Musters ohne Überschreibungen setzt die folgenden Standardwerte:

Amazon SQS-Warteschlange

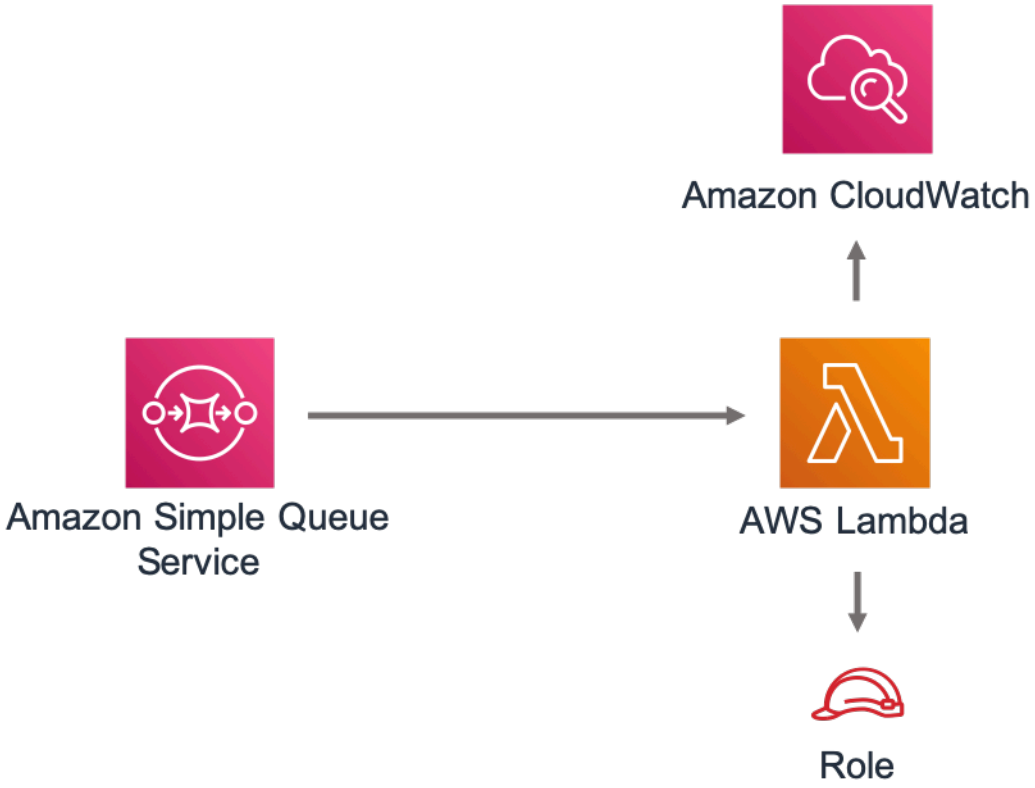
- Bereitstellen der SQS-Warteschlange für unzustellbare Nachrichten für die Quell-Warteschlange
- Aktivieren der serverseitigen Verschlüsselung für Quell-SQS-Warteschlange mithilfe von AWS Managed KMS-Schlüssel
- Erzwingen der Verschlüsselung von Daten während der Übertragung

AWS Lambda-Funktion

- Konfigurieren Sie die IAM-Rolle für den eingeschränkten Zugriffsberechtigungen für Lambda Funktion.
- Aktivieren Sie die Wiederverwendung von Verbindungen mit Keep-Alive für NodeJS Lambda Funktion.

- Aktivieren der X-Ray Nachverfolgung
- Festlegen von Umgebungsvariablen
  - `AWS_NODEJS_CONNECTION_REUSE_ENABLED`(für Node 10.x und höhere Funktionen)

Architecture



GitHub

Um den Code für dieses Muster anzuzeigen, erstellen/anzeigen Probleme und Pull-Anforderungen usw.:

@aws -solutions-constructs/aws-sqs-lambda

core

STABILITY EXPERIMENTAL

Alle Klassen befinden sich in aktiver Entwicklung und unterliegen nicht abwärtskompatiblen Änderungen oder Entfernen in jeder zukünftigen Version. Diese unterliegen nicht derSemantische VersionierungModell. Dies bedeutet, dass Sie möglicherweise Ihren Quellcode aktualisieren müssen, wenn Sie auf eine neuere Version dieses Pakets aktualisieren.

Die Kernbibliothek enthält die grundlegenden Bausteine von AWS Solutions Constructs. Es definiert die Kernklassen, die in den übrigen AWS Lösungskonstrukten verwendet werden.

Standardeigenschaften für AWS CDK-Konstrukte

Die Kernbibliothek legt die Standardeigenschaften für die AWS CDK-Konstrukte fest, die von AWS Solutions Constructs Konstrukts verwendet werden.

Das folgende Beispiel ist das Snippet der Standardeigenschaften für das S3-Bucket-Konstrukt, das von AWS Solutions Constructs erstellt wurde. Standardmäßig aktiviert es die serverseitige Verschlüsselung, Bucket-Versionierung, blockiert den gesamten öffentlichen Zugriff und richtet die S3-Zugriffsprotokollierung ein.

```
{
  encryption: s3.BucketEncryption.S3_MANAGED,
  versioned: true,
  blockPublicAccess: s3.BlockPublicAccess.BLOCK_ALL,
  removalPolicy: RemovalPolicy.RETAIN,
  serverAccessLogsBucket: loggingBucket
}
```

Überschreiben der Standardeigenschaften

Die von der Core-Bibliothek festgelegten Standardeigenschaften können von den vom Benutzer bereitgestellten Eigenschaften überschrieben werden. Der Benutzer kann beispielsweise die Amazon S3 Eigenschaft „Öffentlichen Zugriff blockieren“ überschreiben, um bestimmte Anforderungen zu erfüllen.

```
  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
  },
});
```

Zurücksetzen von Eigenschaftsüberschreibungen

Wenn eine Standardeigenschaft aus der Core-Bibliothek durch eine vom Benutzer bereitgestellte Eigenschaft überschrieben wird, sendet Constructs eine oder mehrere Warnmeldungen an die Konsole aus, in denen die Änderungen hervorgehoben werden. Diese Nachrichten sollen dem Benutzer ein Situationsbewusstsein vermitteln und unbeabsichtigte Überschreibungen verhindern, die Sicherheitsrisiken verursachen könnten. Diese Meldungen werden angezeigt, wenn bereitstellung/build-bezogene Befehle ausgeführt werden, einschließlich`cdk deploy`,`cdk synth`,`npm test`usw.

Beispielnachrichten:`AWS_CONSTRUCTS_WARNING: An override has been provided for the property: BillingMode. Default value: 'PAY_PER_REQUEST'. You provided: 'PROVISIONED'.`

Überschreibungswarnungen umschalten

Warnmeldungen zum Überschreiben sind standardmäßig aktiviert, können jedoch explizit mit der Option`overrideWarningsEnabled`-Shell-Variable.

- ExplizitDeaktivieren vonWarnungen überschreiben, ausführen`export overrideWarningsEnabled=false`.
- ExplizitaktivierenWarnungen überschreiben, ausführen`export overrideWarningsEnabled=true`.

- Führen Sie zum Zurücksetzen des Standard-`unset overrideWarningsEnabled`.

Dokumentversionen

Wenn Sie über -Aktualisierungen von AWS Solutions Constructs benachrichtigt werden möchten, abonnieren Sie den RSS-Feed.

Update-Historie-Änderung	Update-Historie-Beschreibung	Update-Historie-Datum
Aktualisierter Inhalt	Das aws-lambda-ssmstringparameter-Muster wurde hinzugefügt. Andere kleinere Inhaltsaktualisierungen.	27. Mai 2021
Aktualisierter Inhalt	Das aws-lambda-secretsmanager-Muster wurde hinzugefügt. Andere kleinere Inhaltsaktualisierungen.	12. Mai 2021
Aktualisierter Inhalt	Die Eigenschaft wird aktualisiert, um *-Lambda-Muster auszuwählen. Andere kleinere Inhaltsaktualisierungen.	17. April 20
Aktualisierter Inhalt	Es wurde ein Problem in der Walkthrough für Python-Benutzer und aktualisierten Eigenschaftsbeispielen für Konstrukte mit Lambda Funktionen behoben.	30. März 2021
Aktualisierter Inhalt	Kleinere Korrekturen/Aktualisierungen von Musterreq uisiten und Standardeinstellungen für ausgewählte Muster.	8. März 2021
Aktualisierter Inhalt	Kleinere Korrektur/Aktualisierungen von exemplarischen Inhalten.	4. März 2021

Aktualisierter Inhalt	hinzugefügt`aws-lambda-sagemakerendpoint` - Muster und aktualisierte Eigenschaften für ausgewählte Kinesis Firehose-Muster.	24. Februar 2021
Aktualisierter Inhalt	hinzugefügt`aws-kinesisstreams-gluejob` - Muster und aktualisierte Walkthrough-Schritte für Python-Benutzer.	17. Februar 2021
Aktualisierter Inhalt	Aktualisierte Eigenschaften für`aws-cloudfront-*`Muster	9. Februar 2021
Aktualisierter Inhalt	Link zu GitHub für jedes Muster hinzugefügt.	5. Februar 2021
Aktualisierter Inhalt	Aktualisierte Eigenschaften für ausgewählte Muster.	1. Februar 2021
Aktualisierter Inhalt	Aktualisierte Dokumentation der Eigenschaften und Standardeinstellungen für ausgewählte Muster.	4. Januar 2021
Aktualisierter Inhalt	Neue Muster hinzugefügt: aws-cloudfront-mediastore und aws-s3-sqs.	20. Dezember 2020
Aktualisierter Inhalt	AWS-Lambda-Sagemaker Muster entfernt.	17. November 2020

Aktualisierter Inhalt	Neue Muster hinzugefügt: aws-events-rule-kinesisstreams, aws-events-rule-kinesisfirehose-s3 und aws-lambda-sagemaker.	27. Oktober 2020
Aktualisierter Inhalt	Aktualisiert, um brechende Änderungen in den Mustern aws-events-rule-sns und aws-events-rule-sqs zu reflektieren: Klassen- und Schnittstellennamen wurden in pascal case geändert.	22. Oktober 2020
Aktualisierter Inhalt	Es wurden aws-apigateway-sagemakerendpoint und aws-kinesisstreams-kinesisfirehose-s3 Muster hinzugefügt; weitere kleinere Aktualisierungen vorhandener Inhalte.	20. Oktober 2020
Aktualisierter Inhalt	aws-apigateway-iot-Muster hinzugefügt; andere kleinere Aktualisierungen an vorhandenen Inhalten.	7. Oktober 2020
Aktualisierter Inhalt	Minimale bereitstellbare Mustercode-Snippets und Best Practice-Standardwerte für alle Muster wurden aktualisiert.	5. Oktober 2020
Aktualisierter Inhalt	Aktualisierte Eigenschaften für das aws-kinesisstreams-lambda-Muster, um brechende Änderungen zu widerspiegeln.	14. September 2020

Aktualisierter Inhalt	Kleinere Korrektur zum zweiten Teil der exemplarischen Vorgehensweise.	10. September 2020
Aktualisierter Inhalt	Die Muster aws-apigateway-kinesisstreams, aws-events-rule-sns und aws-events-rule-sqs wurden hinzugefügt.	10. September 2020
Aktualisierter Inhalt	Das aws-sns-sqs-Muster wurde hinzugefügt; Aktualisierungen aller SNS-Muster; kleinere typografische Korrekturen.	2. September 2020
Aktualisierter Inhalt	Modulnamen für das aws-sqs-lambda-Muster korrigiert.	31. August 2020
Aktualisierter Inhalt	Der Name des Python-Moduls für das aws-dynamodb-stream-lambda-elasticsearch-kibana-Muster wurde behoben.	31. August 2020
Aktualisierter Inhalt	Aktualisierte Standardwerte für Lambda Muster; andere kleinere Aktualisierungen.	27. August 2020
Aktualisierter Inhalt	Aktualisierte öffentliche Eigenschaften für S3-Muster; aktualisierte Standardwerte für DynamoDB Muster.	10. August 2020
Aktualisierter Inhalt	Mehrere Muster wurden aktualisiert, um die standardmäßige Erzwingung der Verschlüsselung bei der Übertragung hervor	4. August 2020

Aktualisierter Inhalt	Das aws-lambda-sqs-lambda-Muster wurde hinzugefügt; verbesserte Konfigurationsanweisungen im Handbuch Erste Schritte; alle Muster wurden aktualisiert, um zusätzliche Ressourcen über öffentliche Eigenschaften verfügbar zu machen.	27. Juli 2020
Aktualisierter Inhalt	aws-lambda-sqs-Muster hinzugefügt; andere kleinere Updates.	20. Juli 2020
Aktualisierter Inhalt	DeployLambda- und DeployBucket-Eigenschaften aus relevanten Mustern entfernt; andere kleinere Updates.	9. Juli 2020
Aktualisierter Inhalt	Das Muster der aws-lambda-Schritt-Funktion wurde hinzugefügt und kleinere typografische Fehler korrigiert.	7. Juli 2020
Aktualisierter Inhalt	wurde hinzugefügt. -Eigenschaft, um DynamoDB Muster auszuwählen.	25. Juni 2020
Aktualisierter Inhalt	Mehrere Textkorrekturen und Korrekturen für fehlerhafte Links.	23. Juni 2020
Erstversion	AWS Lösungskonstrukte öffentlich verfügbar gemacht.	22. Juni 2020

Notices

Die Kunden sind für ihre eigene unabhängige Bewertung der Informationen in diesem Dokument verantwortlich. Dieses Dokument: (a) dient nur zu Informationszwecken, (b) stellt aktuelle AWS Produktangebote und -praktiken dar, die ohne vorherige Ankündigung geändert werden können und (c) keine Zusagen oder Zusicherungen von AWS und seinen verbundenen Unternehmen, Lieferanten oder Lizenzgebern schafft. AWS Produkte oder -Services werden ohne ausdrückliche oder stillschweigende Gewährleistungen, Zusicherungen oder Bedingungen jeglicher Art bereitgestellt. Die Verantwortung und Haftung von AWS gegenüber seinen Kunden werden durch AWS-Vereinbarungen geregelt. Dieses Dokument gehört, weder ganz noch teilweise, nicht zu den Vereinbarung von AWS mit seinen Kunden und ändert diese Vereinbarungen auch nicht.

© 2020 Amazon Web Services, Inc. oder Tochterfirmen. Alle Rechte vorbehalten.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.