

課題演習 No.3
(CloudFormation + Auto-scaling
+ CloudWatch + SNS)



課題演習4 (CloudFormation + Auto-scaling+ CloudWatch + SNS)

目的:

ある企業でWebサイトを構築する際の検証用と本番用環境に対して、テンプレートを使用して同じ環境での構築を可能にしたいと考えている。今回はその検証用環境を作成したい。

Webページはサイト閲覧時にセキュアな通信で接続できるようにし、さらには運用面でテンプレート環境をベースにしてEC2の自動スケーリングや、状態監視によるアラーム通知といった機能を追加したいと考えている。

1:要件

■以下の要件でインフラを構築してください。

1. 下記①-③を満たす1つの環境を自動構築する。

①2つのAZでパブリックサブネットを作成

②EC2を各パブリックサブネットに設置し、WebサーバはApacheを使用する。

※②のEC2は事前にAMIを作成しておく。

③ALBを使用し、ターゲットグループには各EC2を設定する

(ヒント:CF作成時の「詳細を指定」でスタックの名前をymlファイル名で作成する。

これは2つのymlファイルがスタックの名前で連携されているため。)

-スタックによる構築後-※以下は手動設定(マネージメントコンソール)で行う

2. Webページにはドメインを使用した暗号化されたセキュアな通信をしたい。

3. EC2がCPU負荷率が70%を超えた際に自動でスケールアウト、
負荷率が20%を下回った際にはスケールインできるようにしたい。

4. モニタリングで閾値を超えた際にはアラートを担当者へ
Eメールで発信できるように設定する。

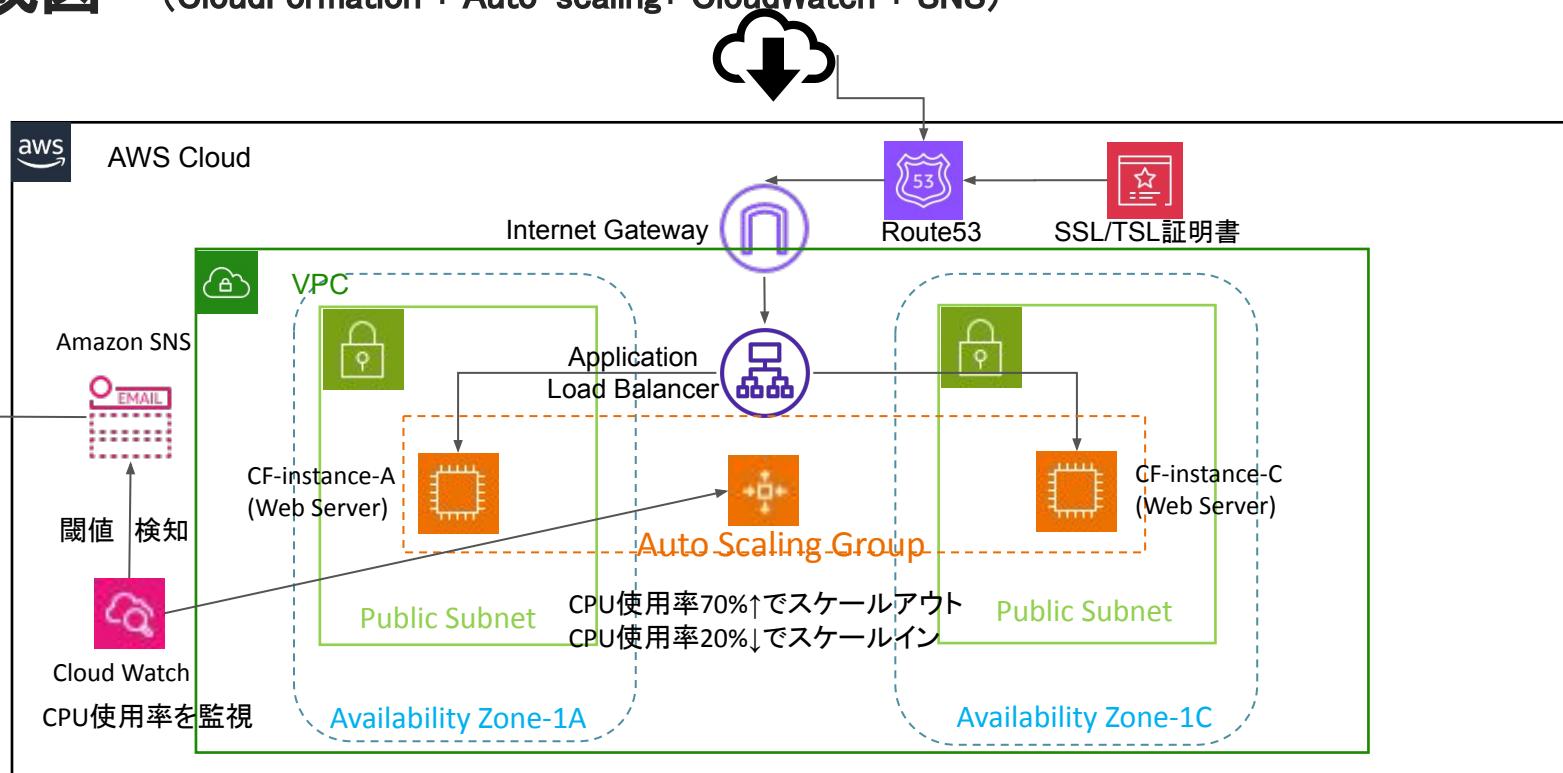
2: 設計

■要件に対しての仕様(解決策)

No	要件	仕様(解決策)
1	下記の条件を満たす1つの環境を自動構築する。 <ul style="list-style-type: none">・2つのAZでパブリックサブネットを作成・EC2を各パブリックサブネットに設置、WebサーバはApacheを使用する。・ALBを使用し、ターゲットグループには各EC2を設定する	EC2インスタンスを作成し、そのインスタンスへapacheサーバーをインストールする。そこからAMIを作成してAMIIDを控えておく。YAMLファイルの中に2つのAZでパブリックサブネットを構築を明示するコードを記載する。控えたAMIIDを使い、EC2をパブリックサブネットの中に構築を明示するコードを記載する。ALBがターゲットグループとして各EC2になるようにコードを記載する。CloudFormationでインフラをIaC化してスタック作成で自動構築。
2	Webページにはドメインを使用した暗号化されたセキュアな通信をしたい。	Route53のホストゾーン内にALBを作成し、そのドメインでALBをエイリアスとして指定する。
3	EC2がCPU負荷率が70%を超えた際に自動でスケールアウト、CPU負荷率が20%を下回った際にはスケールインできるようにしたい。	Auto Scalingを使用して、しきい値を超えた際にはインスタンスの増減できるようにする。
4	モニタリングで閾値を超えた際にはアラートを、担当者へメールで発信できるように設定する。	Cloud WatchでAuto Scaling Groupを監視し、しきい値を超えた際にはCloud Watch Alarmがアラームを発報し、SNSを経由してサブスクリプションで設定して担当者のメールアドレスへメール送信する。

2: 設計

-構成図- (CloudFormation + Auto-scaling+ CloudWatch + SNS)



3: 構築(実装)

①EC2インスタンスの作成

インスタンスを起動 情報

Amazon EC2 では、AWSクラウドで実行される仮想マシン(インスタンス)を作成できます。以下の簡単なステップに従って

名前とタグ 情報

名前
kaldi4-instance さらにタグ

▼ アプリケーションおよび OS イメージ (Amazon マシンイメージ) 情報

AMI は、インスタンスの起動に必要なソフトウェア設定(オペレーティングシステム、アプリケーションセンター、アプ

探しのもののが以下に表示されない場合は、AMI を検索または参照してください。

検索: 同じもののアプリケーションイメージと OS イメージを含むカタログ全体を検索します。

最新 | 自分の AMI | クイックスタート

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Linux | Debian

Amazon マシンイメージ (AMI)

Amazon Linux 2023 AMI
ami-0f72209452eefed4 (64 ビット [x86], uefi-preferred) / ami-08860652e0357ad94f (64 ビット [Arm], uefi)
復元: t2m.1.0 | EBS 有効: true | ルートパーティションタイプ: ext4

説明

Amazon Linux 2023 は、5 年間の長期サポートを備えた、最新の汎用 Linux ベースの OS です。AWS 向けに最適化され

たため実行するための安全で安定した高性能な実行環境を提供するように設計されています。

Amazon Linux 2023 AMI 2023.6.20250218.2 x86_64 HVM kernel-6.1

アーキテクチャ
64 ビット [x86] ▼

ブートモード
uefi-preferred

AMI ID
ami-072298456cc5cb0c4

ユーザー
ec2-user

インスタンスタイプ

t2.micro 無料利用枠の対象

オプション: 1 vCPU | 1 GiB メモリ | 運行世代: true | オンデマンド Windowsベース 料金: 0.0798 USD per Hour
オンドマンド Linuxベース 料金: 0.0152 USD per Hour | オンデマンド Ubuntu Proベース 料金: 0.0715 USD per Hour
オンドマンド Linuxベース 料金: 0.0296 USD per Hour | オンデマンド Linuxベース 料金: 0.0152 USD per Hour

すべての世代

インスタンスタイプを比較

▼ キーヘア (ログイン) 情報

キーヘアを使用してインスタンスに安全に接続できます。インスタンスを起動する前に、選択したキーヘアにアクセスできることを確認してください。

キー名: 必須
test_key_pair ▼ 新しいキーヘアの作成

▼ ネットワーク設定 情報

VPC - 必須 情報

172.31.0.0/16 (デフォルト) ▼

サブネット 情報

VPC: vpc-0123456789012345 | 网关: 172.31.58.47 | 192.168.1.125 | プライベートアドレス: ap-northeast-1d
サブネットのタイプ: アペラビリティーリーン | 私用の既定 IP アドレス: 4091 | CIDR: 172.31.16.0/20

新しいサブネットを作成

パブリック IP の自動割り当て 情報

有効化 ▼

無料利用枠を超過する場合は追加料金が適用されます

ファイアウォール (セキュリティグループ) 情報

セキュリティグループとは、インスタンスのトラフィックを制御する一連のファイアウォールルールです。特定のトラフィックがインスタンスに到達できるようにルールを追加します。

セキュリティグループを作成 ▼ 存在するセキュリティグループを選択する

共通のセキュリティグループ 情報

セキュリティグループを編集 ▼

default sg-08d643e7090c8321 X

VPC: vpc-0123456789012345 | 网关: 172.31.58.47 | 192.168.1.125

セキュリティグループのルールを追加

ここでは既に既存のルール (セキュリティグループ) が選択されています。すべてのネットワークインターフェイスに適用。またはすべてのネットワークインターフェイスから削除されます。

▼ 高速なネットワーク設定

3:構築(実装)

①作成をしたインスタンスへSSH接続しApacheをインストール

```
C:\Users\edu>ssh -i "C:\Users\edu\Downloads\test_key_pair.pem" ec2-user@35.77.35.116
The authenticity of host '35.77.35.116 (35.77.35.116)' can't be established.
ED25519 key fingerprint is SHA256:HalMay/A+NyC6m1rhZt07oxzdEkvrres9HuJMHu/cK8.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '35.77.35.116' (ED25519) to the list of known hosts.

[Amazon Linux 2023]
[https://aws.amazon.com/linux/amazon-linux-2023]
[Dependencies resolved.]
[httpd]
[httpd-filesystem]
[httpd-tools]
[libbrotli]
[mailcap]
[apr-util-openssl]
[mod_http2]
[mod_lua]

Transaction Summary
```

3: 構築(実装)

①Webサーバーのファイル内を編集するコマンドを入力

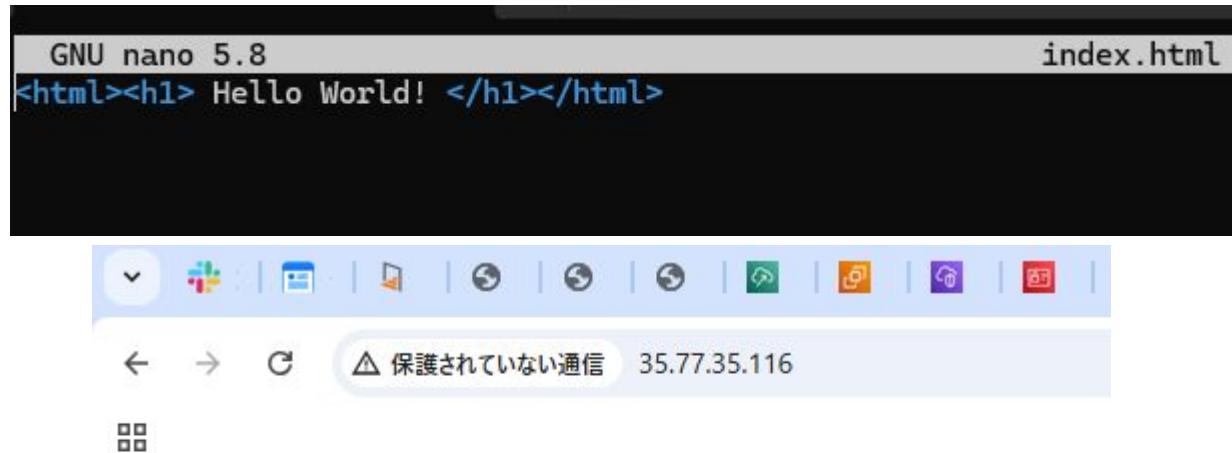
```
Verifying .mod_lua-2.4.62-1.amzn2023.x86_64

Installed:
  apr-1.7.5-1.amzn2023.0.4.x86_64
  apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
  httpd-2.4.62-1.amzn2023.x86_64
  httpd-filesystem-2.4.62-1.amzn2023.noarch
  libbrotli-1.0.9-4.amzn2023.0.2.x86_64
  mod_http2-2.0.27-1.amzn2023.0.3.x86_64
  mod_lua-2.4.62-1.amzn2023.x86_64
  apr-util-1.6.3-1.amzn2023.0.1.x86_64
  generic-logos-httpd-18.0.0-12.amzn2023.0
  httpd-core-2.4.62-1.amzn2023.x86_64
  httpd-tools-2.4.62-1.amzn2023.x86_64
  mailcap-2.1.49-3.amzn2023.0.3.noarch
  mod_lua-2.4.62-1.amzn2023.x86_64

Complete!
[root@ip-172-31-27-140 ec2-user]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/
[root@ip-172-31-27-140 ec2-user]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
    Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
    Active: inactive (dead)
      Docs: man:httpd.service(8)
[root@ip-172-31-27-140 ec2-user]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
    Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
    Active: inactive (dead)
      Docs: man:httpd.service(8)
[root@ip-172-31-27-140 ec2-user]# cd /var/www/html
[root@ip-172-31-27-140 html]# nano index.html
[root@ip-172-31-27-140 html]# systemctl enable httpd
[root@ip-172-31-27-140 html]# systemctl restart httpd
```

3: 構築(実装)

- ①HTMLでHello Worldを記述、保存、再起動



Hello World!

3: 構築(実装)

②作成したインスタンスからAMIを作成し、AMIIDをYAMLファイルへ記述

イメージを作成 情報

イメージ(AMIと名づけます)は、EC2 インスタンスの起動時に適用され

インスタンス ID
 i-0fc42d51cd3a90d8b (kadal4-instance)

イメージ名
kadal4-ami

最大 128 文字。作成後に変更することはできません。

イメージの説明 - オプション
イメージの説明
最大 255 文字

インスタンスを再起動

選択すると、Amazon EC2 はインスタンスを再起動して、アサインされたボリューム

インスタンスポリューム

ストレージタイプ デバイス スナップショット

EBS /dev/xvda ボリュームから新しいスナ

```
KeyName:  
  Description : Name of an existing EC2 KeyPair  
  Type: AWS::EC2::KeyPair::KeyName  
  ConstraintDescription : Can contain only A-Z  
  
Resources:  
  MyEC2Instance1:  
    Type: AWS::EC2::Instance  
    Properties:  
      ImageId: !Ref ImageId  
      InstanceType: t2.micro  
      SubnetId: !ImportValue VPC-koutiku-Public  
      BlockDeviceMappings:  
        -  
          DeviceName: /dev/xvda  
          Ebs:  
            VolumeType: gp2  
            VolumeSize: 8  
          Tags:  
            - Key: Name  
              Value: CF-public-1a  
          KeyName: !Ref KeyName  
          SecurityGroupIds:  
            - !GetAtt "InstanceSecurityGroup.Group  
  
  MyEC2Instance2:  
    Type: AWS::EC2::Instance  
    Properties:  
      ImageId: !Ref ImageId  
      InstanceType: t2.micro  
      SubnetId: !ImportValue VPC-koutiku-Public  
      BlockDeviceMappings:  
        -
```

3: 構築(実装)

②Route53でホストゾーンを作成

三 Route 53 > ホストゾーン > ホストゾーンの作成

ホストゾーンの作成 情報

ホストゾーン設定

ホストゾーンは、example.comなどのドメインおよび
ドメイン名 | 情報
これは、トラフィックをルーティングするドメインの名前です。
[入力欄] 有効な文字:a~z、0~9、!"#\$%&'()*+、-/;:<=>?@[
説明 - オプション | 情報
この値で、同じ名前のホストゾーンを区別できます。
ホストゾーンは次の目的で使用されます。
説明は最大 256 文字です。0/256

タイプ | 情報

このタイプは、インターネットまたは Amazon VPC でトラフィ

レコード (5) DNSSEC署名 ホストゾーンのタグ (0)

レコード (5) 情報 utomatic モードは最適なフィルタ結果に最適化された現在の検索動作です。モードを変更するには、[設定] に移動します。

Q プロパティまたは値でレコードをフィルタリングする タイプ ▾ ルーティング規則 ▾ エイリアス ▾ 1 / 1 ヘルプ

レコード名	タ... ▾	ルーティング規則	差別化	エイリアス	値/トラフィックのルーティング	TTL (秒)	ヘルプ
a02.ope-edu...	NS	シンプル	-	いいえ	ns-1293.awsdns-33.org. ns-509.awsdns-63.com. ns-748.awsdns-29.net. ns-1997.awsdns-57.co.uk.	172800	-
a02.ope-edu...	SOA	シンプル	-	いいえ	ns-1293.awsdns-33.org. aws...	900	-
_9d265d5c06...	CNAME	シンプル	-	いいえ	_180e8d4fde32d72ba68234...	300	-
www.a02.ope...	A	シンプル	-	いいえ	54.249.59.40	300	-
_1b7718bfe1...	CNAME	シンプル	-	いいえ	_342b2428e49f03f31ba695...	300	-

3: 構築(実装)

②作成したホストゾーン内のAレコードの作成、エイリアスでALBを指定

レコードを編集

レコード名 | 情報
www .a02.ope-education.link

レコードタイプ | 情報
A – IPv4 アドレスと一部の AW... ▾

ルートドメインのレコードを作成するには、空白のままにします。

エイリアス

トラフィックのルーティング先 | 情報
Application Load Balancer と Classic Load Balancer へのエイリアス ▾
アジアパシフィック (東京) ▾

Q dualstack.CF-elb-1.████████.1.elb.amazonaws.com X

エイリアスホストゾーン ID: Z14GRHDCWA56QT

ルーティングポリシー | 情報
シンプルルーティング ▾

ターゲットのヘルスを評価
 いいえ

キャンセル 保存



3: 構築(実装)

③Route53で作成したホストゾーンを使いACMで証明書を発行し、YAMLファイルのALBのリスナーへ証明書ARNを記述

パブリック証明書をリクエスト

ドメイン名
証明書の 1 つ以上のドメイン名を指定します。
完全修飾ドメイン名 | [情報](#)

この証明書に別の名前を追加
この証明書にはさらに名前を追加できます。例: www.example.com の証明書をリクエストする場合、影響がいかずか

検証方法 | [情報](#)
ドメインの所有権を検証する方法を選択
 DNS 検証 - 離脱
証明書リクエストでドメインの DNS 設定を変更できる場合は、このオプションを選択します。
 E メール検証
証明書リクエストでドメインの DNS 設定を変更する許可がない場合、または当該許可を得難い場合は、このオプション

キーアルゴリズム | [情報](#)
暗号化アルゴリズムを選択します。一部のアルゴリズムは、一部の AWS サービスでサポートされていない場合、
 RSA 2048
RSA は、最も広く使用されているキータイプです。
 ECDSA P-256
暗号化強度は RSA 512 と同等です。
 ECDSA P-384
暗号化強度は RSA 7680 と同等です。

タグ | [情報](#)
リソースに関連付けられたタグがありません。

Port: 80
Protocol: HTTP

ELBLListenerHTTPS:
Type: AWS::ElasticLoadBalancingV2::Listener
Properties:
Certificates:
- CertificateArn: arn:aws:acm:ap-northeast-1:5[REDACTED]
DefaultActions:
- TargetGroupArn: !Ref ELBtargetGP
Type: forward
LoadBalancerArn: !Ref InternetELB
Port: 443
Protocol: HTTPS

3: 構築(実装)

④ YAMLファイルをCloudFormationでスタックの作成をし、インフラを展開

スタックの作成

前提条件 - テンプレートの準備

AWS ジェネレーターで既存のリソースをスキャンしてテンプレートを作成することができます。

テンプレートの準備

各スタックはテンプレートに基づきます。テンプレートとは、スタックに渡る AWS リソースに関する設定情報を含む JSON 形

既存のテンプレートを選択

既存のテンプレートをアップロードまたは選択します。

テンプレートの指定

この GitHub リポジトリには、新しいインフラストラクチャプロジェクトの開始に役立つサンプル CloudF

テンプレートソース

テンプレートを選択すると、保存先となる Amazon S3 URL が生成されます。テンプレートは、スタックのリソースおよ

Amazon S3 URL

テンプレートに Amazon S3 URL を指定します。

テンプレートファイルのアップロード

テンプレートをコンソールに直接アップロードします。

テンプレートファイルのアップロード

ファイルの選択

VPC-koutiku.yml

JSON または YAML 形式のファイル

スタックの作成

前提条件 - テンプレートの準備

AWS ジェネレーターで既存のリソースをスキャンしてテンプレートを作成することもできます。

テンプレートの準備

各スタックはテンプレートに基づきます。テンプレートとは、スタックに渡る AWS リソースに関する設定情報を含む

既存のテンプレートを選択

既存のテンプレートをアップロードまたは選択します。

スタック (2)

Q スタック名によるフィルター

ステータスのフィルター

アクティブ

ネストされているものを表示

< 1 >

スタック

CF-instance

2025-02-26 15:13:52 UTC+0900

CREATE_COMPLETE

VPC-koutiku

2025-02-26 15:08:55 UTC+0900

CREATE_COMPLETE

テンプレートの指定

この GitHub リポジトリには、新しいインフラストラクチャプロジェクトの開始に役立つサンプル CloudF

テンプレートソース

テンプレートを選択すると、保存先となる Amazon S3 URL が生成されます。テンプレートは、スタックのリソースおよ

Amazon S3 URL

テンプレートに Amazon S3 URL を指定します。

テンプレートファイルのアップロード

テンプレートをコンソールに直接アップロードします。

テンプレートファイルのアップロード

ファイルの選択

CF-instance.yml

JSON または YAML 形式のファイル

S3 URL: <https://s3.ap-northeast-1.amazonaws.com/cf-templates-1dkrok07f695q-ap-northeast-1/2>

3: 構築(実装)

⑤AutoScalingで使う起動テンプレートの作成

The screenshot shows the AWS CloudFormation console interface. The top navigation bar includes links for Lambda, API Gateway, CloudFormation, Lightsail, WAF & Shield, Route 53, AWS Application Migration Service, and CloudWatch Metrics. Below the navigation, there's a message: '起動テンプレートを作成' (Create Launch Configuration). A note below it says: '起動テンプレートを作成することで、後で再利用、共有、起動できる保存済みインスタンス設定を作成できます。テンプレートには複数の' (You can create a launch configuration for reuse, sharing, and starting instances later. A template can contain multiple...). The main form is titled '起動テンプレート名と説明' (Launch Configuration Name and Description). It has two input fields: '起動テンプレート名 - 必須' (Launch Configuration Name - Required) containing 'kada14-template', and 'テンプレートバージョンの説明' (Description of the template version) containing 'MyApp 用の 本番ウェブサーバー'. Below these fields is a note: 'このアカウントに既存である必要があります。最大 255 文字。スペースや、『&』、『!』、『*』、『@』などの特殊文字は使用できません。' (This account must already exist. Maximum 255 characters. Spaces and special characters like &, !, *, @ are not allowed.). At the bottom of the form, there's a section titled 'Auto Scaling のガイダンス | 情報' (Auto Scaling Guidance | Information) with a note: 'EC2 Auto Scaling でこのテンプレートを使用する場合は、これを選択します' (If you plan to use this template with EC2 Auto Scaling, select this). A checkbox labeled 'EC2 Auto Scaling で使用できるテンプレートをセットアップする際に役立つガイダンスを提供' (Provide guidance useful for setting up this template when using it with EC2 Auto Scaling) is checked. At the very bottom, there are two buttons: 'テンプレートタグ' (Template Tags) and 'ソーステンプレート' (Source Template).

3: 構築(実装)

⑤起動テンプレートの続きの設定

起動テンプレートのコンテンツ

起動テンプレートの詳細を以下で説明します。フィールドを空白のままにすると、フィールドが起動テンプレートに含まれません。

▼ アプリケーションおよび OS イメージ (Amazon マシンイメージ) [情報](#)

AMI は、インスタンスの起動に必要なソフトウェア層(オペレーティングシステム、アプリケーションサーバー、アプリケーション等)を含む。AMI を候補または選択してください。

Q 何千ものアプリケーションイメージと OS イメージを含むカタログ全体を検索します。

最新 [自分の AMI](#) クイックスタート

起動テンプレートの段階に含めない 自己所有 自分と共有

Amazon マシンイメージ (AMI)

kadai4-ami
ami-02cdcbd19b135d29254
2023-02-28T05:52z03.000Z 値付け: hvm ENA 対応: true ルートデバイスタイプ: vols フロートモード: usd-preferred

説明

キー名: [x86_64](#) AMI ID: ami-02cdcbd19b133d29254

▼ インスタンスタイプ [情報](#) | [アドバイスを要ける](#)

インスタンスタイプ

t2.micro
フロントエンド: 1 vCPU, 1 GiB メモリ, 順行世代: true オンデマンド Windowsベース 料金: 0.0198 USD per Hour
オンデマンド SUSEベース 料金: 0.0152 USD per Hour オンデマンド Ubuntu Proベース 料金: 0.017 USD per Hour
オンデマンド RHELベース 料金: 0.0296 USD per Hour オンデマンド Linuxベース 料金: 0.0152 USD per Hour

ソフトウェアがプリインストールされた AMI には追加料金がかかります

▼ インスタンスタイプ [情報](#) | [アドバイスを要ける](#)

インスタンスタイプ

t2.micro
フロントエンド: 1 vCPU, 1 GiB メモリ, 順行世代: true オンデマンド Windowsベース 料金: 0.0198 USD per Hour
オンデマンド SUSEベース 料金: 0.0152 USD per Hour オンデマンド Ubuntu Proベース 料金: 0.017 USD per Hour
オンデマンド RHELベース 料金: 0.0296 USD per Hour オンデマンド Linuxベース 料金: 0.0152 USD per Hour

ソフトウェアがプリインストールされた AMI には追加料金がかかります

▼ キーペア (ログイン) [情報](#)

キーペアを使用してインスタンスに安全に接続できます。インスタンスを起動する前に、選択してください。

キーペア名: test_key_pair

▼ ネットワーク設定 [情報](#)

サブネット | [情報](#)

VPC: vpc-0f8de0814950678851 所属 subnet: 10.0.79.198/0.25 アベイラビリティゾーン: ap-northeast-1a ジーンのタイプ: アベイラビリティゾーン 料金単位なし IP アドレス: 249 CIDR: 10.0.0.0/24

セキュリティグループを割り当てる場合、ネットワーキングインターフェイスがテンプレートに自動的に追加されます。

ファイアウォール (セキュリティグループ) | [情報](#)

セキュリティグループとは、インスタンスのトラフィックを制限する一連のファイアウォールルールです。特記

現存のセキュリティグループを選択する セキュリティグループを

共通のセキュリティグループ | [情報](#)

セキュリティグループを編集

CF-instance-InstanceSecurityGroup [REDACTED] X
VPC: vpc-0f8de0814950678851

ここに追加または削除したセキュリティグループは、すべてのネットワーキングインターフェイスに追加。または、

▶ 高速なネットワーク設定

3: 構築(実装)

⑥AutoScaling の作成、サブネット2つ選択、ALBにアタッチ

起動テンプレートまたは起動設定を選択する [Info](#)
この Auto Scaling グループによって起動されたすべての EC2 インスタンスに共通の設定が含まれる起動テンプレートを選択してください。

名前
Auto Scaling グループ名
グループ名を入力してください。
kada14_AutoScaling
複数のリージョンにあるこのカウントに異なります。

起動テンプレート [Info](#)
起動テンプレート
Amazon マシンイメージ (AMI), インスタンスタイプ, キーペア, セキュリティグループなど。インスタンスレベルの kada14-template
起動テンプレートを作成する [\[+\]](#)
バージョン
Default (1) [\[C\]](#)
起動テンプレートバージョンを作成する [\[+\]](#)
説明
-
AMI ID
ami-02cbd19b133d29254
キーペア名
test_key_pair
セキュリティグループ
-
セキュリティグループ ID
[REDACTED]
ストレージ (ボリューム)
-
追加の詳細
-
作成日
Wed Feb 26 2025 15:38:34 GMT+0900 (G)

キーペア名
test_key_pair
セキュリティグループ ID
sg-0baa007e12050bcc2 [\[C\]](#)

追加の詳細
ストレージ (ボリューム)
-
作成日
Wed Feb 26 2025 15:38:34 GMT+0900 (GMT+09:00)

他のサービスと統合する - 省略可能 [Info](#)
コードバランサーを使用して、ネットワークトラフィックを複数のサーバーに分散します。VPC Lattice でサービス間通信を可能にします。ノードを渡さることができます。また、ヘルスチェックの選択やモニタリングをカスタマイズすることもできます。

ロードバランシング [Info](#)
以下のオプションを使用して、Auto Scaling グループを既存のロードバランサー、または定義した新しいロードバランサーにアタッチすることができます。

ロードバランサーがありません
Auto Scaling グループへのトラフィックが、ロードバランサーに先立つことはありません。

既存のロードバランサーにアタッチする
既存のロードバランサーから選択します。

既存のロードバランサーにアタッチする
Auto Scaling グループにアタッチするロードバランサーを選択します。

ロードバランサーのターゲットグループから選択する
このオプションは、Application Load Balancer、Network Load Balancer、または Gateway Load Balancer をアタッチできます。

Classic Load Balancer から

既存のロードバランサーターゲットグループ
Auto Scaling グループと同じ VPC に関するインスタンスのターゲットグループのみを選択できます。

ターゲットグループを選択する
ELBtargetGP | HTTP [\[C\]](#)
Application Load Balancer: CF-elb

アベイラビリティゾーンのディストリビューション - 新規
Auto Scaling は、アベイラビリティゾーン間で自動的にインスタンスのバランスを取ります。あるゾーンで初期に失敗した場合は、他のを試してください。

パラシング (ベストエフォート)
あるアベイラビリティゾーンで初期が失敗した場合、Auto Scaling は別の正常なアベイラビリティゾーンで初期を試みます。

パラシング
あるアベイラビリティゾーンで初期が失敗した場合、Auto Scaling は同じアベイラビリティゾーンで初期を試みます。

3: 構築(実装)

⑥AutoScalingのグループサイズとスケーリングを設定

グループサイズとスケーリングを設定する - 省略可能 [Info](#)
グループの希望する容量とスケーリング制限を指定します。グループのサイズを調整するために、オプションで

グループサイズ [Info](#)
Auto Scaling グループの初期サイズを設定します。グループを作成したあと、手動または自動スケーリング。

希望する容量タイプ
希望する容量の割り当て方を選択します。vCPU とメモリ (GB) は、一括のインスタンス属性で構成された混合インスタンス。

希望するキャパシティ
グループサイズを設定してください。

スケーリング [Info](#)
Auto Scaling グループのサイズは、需要の変化に合わせて手動または自動で変更できます。

スケーリング制限
希望する容量をどれだけ削減できるかを設定します。

最小の希望する容量 **最大の希望する容量**

手動スケーリング - 省略可能
ターゲットの追跡ポリシーを使用するかどうかを選択する [Info](#)
Auto Scaling グループを作成した後に、他のメトリクスベースのスケーリングポリシーとスケジュールされたスケーリング。

スケーリングポリシーなし
Auto Scaling グループは初期サイズのままとなり、需要に合わせて自動的にサイズ変更されることはありません。

インスタンスマネナンスポリシー [Info](#)
インスタンスを引き抜きイベント中の Auto Scaling グループの可用性を制限します。これには、ヘルスチェックを複数回ために自動的に行われるイベント (リバランシングイベントと呼ばれる) が含まれます。

可用性の要件状況に応じて引き抜き方法を選択してください

行動の選択
 ポリシーなし
リバランス引き抜きイベントの場合、新しいインスタンスは他のインスタンスが終了する前に起動します。そのためのイベントでは、インスタンスは終了します。

可用性を優先
 終了する前に起動
新しいインスタンスが終了するまで待ってから、他のインスタンスを終了します。これにより、希望する容量を上回ることがあります。

コロナ

その他のキャパシティ設定

キャパシティ予約の詳細設定 [Info](#)
Auto Scaling で既存のキャパシティ予約またはキャパシティ予約リソースグループのどちらにインス

デフォルト
Auto Scaling は、起動テンプレートのキャパシティ予約設定を使用します。

なし
インスタンスはキャパシティ予約には認証されません。

キャパシティ予約のみ
インスタンスはキャパシティ予約のみに認証されます。キャパシティが利用できない場合、イン

最初にキャパシティ予約
インスタンスは最初にキャパシティ予約に認証しようとします。キャパシティが利用できない場合

その他の設定

インスタンスのスケールイン保護
スケールインからの保護が有効になっている場合、新しく起動されたインスタンスはデフォルトです。
 インスタンスのスケールイン保護を有効にする

モニタリング [Info](#)
 CloudWatch 内でグループメトリクスの収集を有効にする

デフォルトのインスタンスのウォームアップ [Info](#)
使用状況データがまだ強制されないため、新しいインスタンスの CloudWatch メトリクスがグループ。

デフォルトのインスタンスのウォームアップを有効にする

3: 構築(実装)

⑥AutoScalingの動的スケーリングポリシーを70%以上で1インスタンスを追加と20%以下で1インスタンスを削除の設定

The screenshot shows the AWS CloudWatch Metrics Insights interface for creating a dynamic scaling policy. It is divided into two main sections: '動的スケーリングポリシーを作成する' (Create Dynamic Scaling Policy) on the left and '動的スケーリングポリシーを作成する' (Create Dynamic Scaling Policy) on the right.

Left Panel (Create Dynamic Scaling Policy):

- ポリシータイプ:** ステップスケーリング
- スケーリングポリシー名:** CPU_Add
- CloudWatch アラーム:** ScaleOutAlarm - CF
- アクションを実行する:** 増加 (Add)
 - 1 | 対象ユニット | 次のとき: 70 | >= CPUUtilization < 加盟大
- ステップを追加する**
- インスタンスのウォームアップ:** 300 秒

Right Panel (Create Dynamic Scaling Policy):

- ポリシータイプ:** ステップスケーリング
- スケーリングポリシー名:** CPU_Remove
- CloudWatch アラーム:** ScaleInAlarm - CF
- アクションを実行する:** 減少 (Remove)
 - 1 | 対象ユニット | 次のとき: 20 | >= CPUUtilization > 加盟大

Bottom Navigation:

- 詳細 | 統合・新規 | オートスケーリング | **インスタンス管理** | インスタンスの更新 | アクティビティ | モニタリング

Instances (2) Table:

インスタンス ID	ライフサイクル	インスタンスタイプ	重み付き容量	起動テンプレート	アベイラビリティゾーン	ヘルスステータス	以下のものからの...
[REDACTED]	InService	t2.micro	-	kada14-template [REDACTED]	ap-northeast-1a	Healthy	
[REDACTED]	InService	t2.micro	-	kada14-template [REDACTED]	ap-northeast-1a	Healthy	

3: 構築(実装)

⑦SNSでスケールアウト、インのトピックを作成

① 新機能
Amazon SNS はハイスクループット FIFO トピックをサポートするようになりました。詳細は[こちら](#)。

トピックの作成

詳細

タイプ | **情報**
トピックの作成後にトピックタイプを変更することはできません。

FIFO (先入れ先出し、先出し)

- 画面に保存されたメッセージの順序付け
- 1回のみメッセージ配信
- サブスクリプションプロトコル: SQS, Lambda, Data Firehose, HTTP, SMS, Eメール、モバイルアプリケーションエンドポイント

スタンダード

- ベストエフォート型メッセージの順序付け
- 少なくとも1回のメッセージ配信
- サブスクリプションプロトコル: SQS, Lambda, Data Firehose, HTTP, SMS, Eメール、モバイルアプリケーションエンドポイント

名前
ScaleInAlarm-SNS

最大文字数は 256 文字です。英数字、ハイフン [-]、およびアンダースコア [_] を含めることができます。

表示名 - オプション | **情報**
SMS のサブスクリプションでこのトピックを使用するには、表示名を入力します。SMS メッセージには最初の 10 文字のみが表示されます。

My Topic

最大 100 文字。

▶ 暗号化 - オプション
Amazon SNS はデフォルトで、転送中に暗号化を行います。サーバー側の暗号化を有効にすると、保存中に行った暗号化をトピックに適用します。

Amazon SNS はハイスクループット FIFO トピックをサポートするようになりました。詳細は[こちら](#)。

トピックの作成

詳細

タイプ | **情報**
トピックの作成後にトピックタイプを変更することはできません。

FIFO (先入れ先出し、先出し)

- 画面に保存されたメッセージの順序付け
- 1回のみメッセージ配信
- サブスクリプションプロトコル: SQS

スタンダード

- ベストエフォート型メッセージの順序付け
- 少なくとも1回のメッセージ配信
- サブスクリプションプロトコル: SQS, Lambda, Data Firehose, HTTP, SMS, Eメール、モバイルアプリケーションエンドポイント

名前
ScaleOutAlarm-SNS

最大文字数は 256 文字です。英数字、ハイフン [-]、およびアンダースコア [_] を含めることができます。

表示名 - オプション | **情報**
SMS のサブスクリプションでこのトピックを使用するには、表示名を入力します。SMS メッセージには最初の 10 文字のみが表示されます。

My Topic

最大 100 文字。

▶ 暗号化 - オプション
Amazon SNS はデフォルトで、転送中に暗号化を行います。サーバー側の暗号化を有効にすると、保存中に行った暗号化をトピックに適用します。

3: 構築(実装)

⑦作成したトピックからサブスクリプションを作成、送信先のメールアドレスを指定

Amazon SNS はハイスループット FIFO トピックをサポートするようになりました。詳細はこちら [\[↗\]](#)

サブスクリプションの作成

詳細

トピック ARN

プロトコル
サブスクリーブするエンドポイントのタイプ

エンドポイント
Amazon SNS から通知を受け取る E メールアドレス

① サブスクリプションを作成した後、それを確認する必要があります。 [情報](#)

▶ サブスクリプションフィルターポリシー - オプション [情報](#)
このポリシーで受信者が受け取るメッセージがフィルター処理されます。

▶ Redrive ポリシー (デッドレターキュー) - オプション [情報](#)
配信不能メッセージをデッドレターキューに送信します。

3: 構築(実装)

⑧CloudWatchAlarmの設定



3: 構築(実装)

⑧メトリクス選択で関連付けるAutoScalingGroupを指定しCPU使用率のメトリクスを選択



3: 構築(実装)

⑧メトリクスのしきい値を70%以上と20%以下でそれぞれ設定



ストリクス

グラフ
このアラームは青線が1分 内の1データポイントで下回る み線を超える場合に、トリガられます。

Percent

70

56.6

52.7

15:00 15:30 16:00 16:30 17:00 17:30

CPUUtilization

名前空間
AWS/EC2

メトリクス名
CPUUtilization

AutoScalingGroupName
kada14-AutoScaling

統計
平均値

期間
1分

条件

しきい値の種類

静的
値をしきい値として使用

异常検出
バンドをしきい値として使用

CPUUtilization が次の時
アラームを作成を実行します。

より大きい
× 1.5倍

以上
× 1.5倍

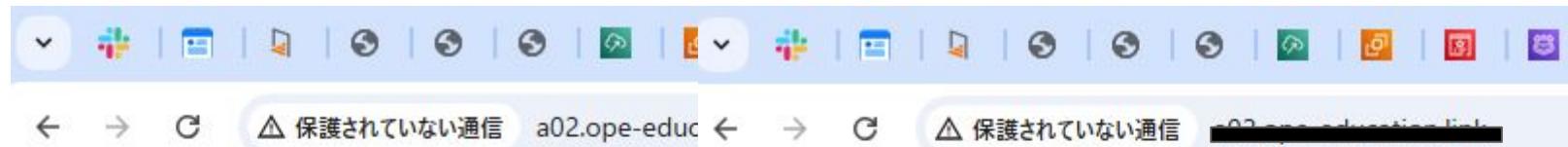
以下
× 1.5倍

...よりも
しきい値を実行します。

20

4: テスト結果(検証)

- ①起動している二つのインスタンスへSSH接続し、Webサーバーファイル内を識別できるように1Aと1Cを加えて編集し、ALBでトラフィックの負荷分散を出来ているか確認



Hello World! 1A Hello World! 1C

4: テスト結果(検証)

②AutoScalingとCloudWatchAlarmが機能しているか確認のため、EC2インスタンスへストレスかけ

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '52.193.199.136' (ED25519) to the list of known hosts.

      _#
      \###_          Amazon Linux 2023
      \###\_
      \##|          https://aws.amazon.com/linux/amazon-linux-2023
      \#/ __ V~ ' ->
      / 
      /_/
      /_/
      /m/' 

Last login: Wed Feb 26 05:31:40 2025 from 133.32.128.50
[ec2-user@ip-10-0-0-144 ~]$ wget https://rpmfind.net/linux/dag/redhat/el7/en/x86_64/dag/RPMS/stress-1.0.2-1.el7.rf.x86_64.rpm
--2025-02-26 08:50:33--  https://rpmfind.net/linux/dag/redhat/el7/en/x86_64/dag/RPMS/stress-1.0.2-1.el7.rf.x86_64.rpm
Resolving rpmfind.net (rpmfind.net)... 195.220.108.108
Connecting to rpmfind.net (rpmfind.net)|195.220.108.108|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 36804 (36K) [application/x-rpm]
Saving to: 'stress-1.0.2-1.el7.rf.x86_64.rpm'

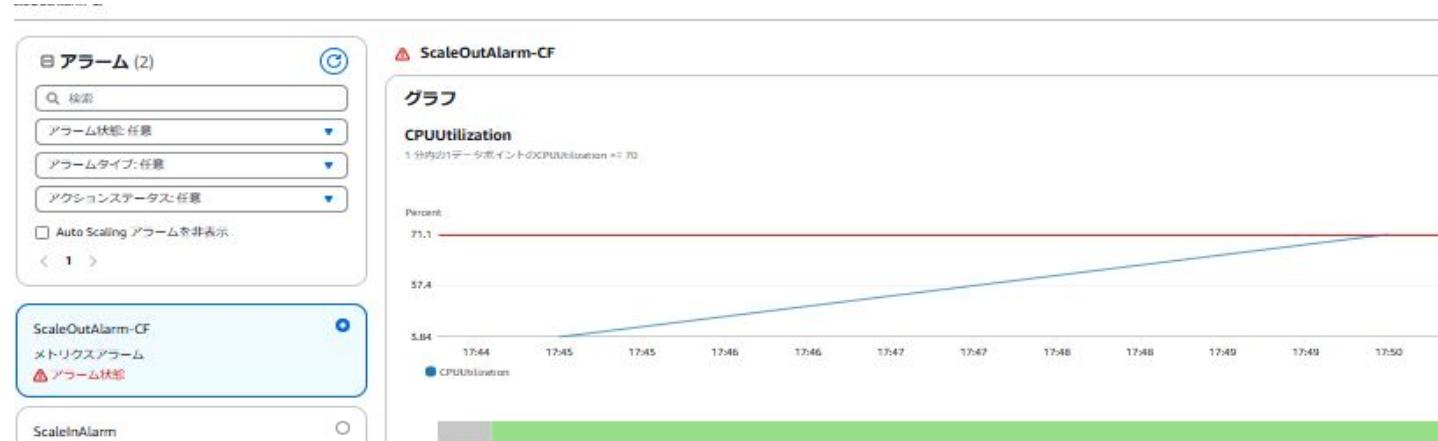
stress-1.0.2-1.el7.rf.x86_64. 100%[=====] 35.94K  151KB/s   in 0.2s

2025-02-26 08:50:34 (151 KB/s) - 'stress-1.0.2-1.el7.rf.x86_64.rpm' saved [36804/36804]

[ec2-user@ip-10-0-0-144 ~]$ sudo rpm -ivh stress-1.0.2-1.el7.rf.x86_64.rpm
warning: stress-1.0.2-1.el7.rf.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 6b8d79e6: NOKEY
Verifying... #####
Preparing... #####
Updating / installing...
 1:stress-1.0.2-1.el7.rf #####
[ec2-user@ip-10-0-0-144 ~]$ rpm -qa|grep stress
stress-1.0.2-1.el7.rf.x86_64
[ec2-user@ip-10-0-0-144 ~]$ stress -c 1 -q &
[1] 8249
```

4: テスト結果(検証)

③ CloudWatchAlarmで70%を超えていること確認



4: テスト結果(検証)

④負荷掛け後にインスタンスが各AZ1つずつから2つずつにスケールアウトしていることを確認

インスタンス (4)							アクション
インスタンスをフィルタリングする							アクション
インスタンス ID	ライフサイクル	インスタンスタイプ	割り当て容量	起動テンプレート	アベイラビリティゾーン	ヘルステータス	以下のものからの選択
[REDACTED]	InService	t2.micro	-	kada14-template [バージョン: 1] /バージョン: 1	ap-northeast-1a	Healthy	
[REDACTED]	InService	t2.micro	-	kada14-template [バージョン: 1] /バージョン: 1	ap-northeast-1c	Healthy	
[REDACTED]	InService	t2.micro	-	kada14-template [バージョン: 1] /バージョン: 1	ap-northeast-1a	Healthy	
[REDACTED]	InService	t2.micro	-	kada14-template [バージョン: 1] /バージョン: 1	ap-northeast-1c	Healthy	

ライフサイクルフック (0) <small>Info</small>	アクション	ライフサイクルフックを作成する
------------------------------------	-------	-----------------

4: テスト結果(検証)

⑤負荷掛けコマンド終了後CPU使用率が20%を下回るのを確認



4: テスト結果(検証)

⑥CPU使用率が20%を下回ったあとインスタンスが各AZ1つずつに戻っていくことを確認

The screenshot shows the AWS Lambda Instances Management interface. The top navigation bar includes tabs for 詳細, 統合・新規, オートスケーリング, **インスタンス管理**, インスタンスの更新, アクティビティ, and モニタリング. The **インスタンス管理** tab is currently selected.

The main section is titled "インスタンス (2)". It displays two instances:

インスタンス ID	ライフサイクル	インスタンスタイプ	重み付き容量	起動テンプレート	アベイラビリティゾーン	ヘルスステータス
[REDACTED]	InService	t2.micro	-	kada4-template	ap-northeast-1a	Healthy
[REDACTED]	InService	t2.micro	-	kada4-template	ap-northeast-1c	Healthy

Below the instances, there is a section titled "ライフサイクルフック (0)" with a "Lifecycle Hookを作成する" button.

4: テスト結果(検証)

⑦スケールアウト、インのアクションをした内容をSNSからメール通知されていることを確認

件名	送信者	日時	サイズ
ALARM: "ScaleOutAlarm-CF" in Asia Pacific (Tokyo)	AWS Notifications	09:04:36	5.46K
ALARM: "ScaleOutAlarm-CF" in Asia Pacific (Tokyo)	AWS Notifications	08:51:36	5.46K
ALARM: "ScaleInAlarm" in Asia Pacific (Tokyo)	AWS Notifications	08:44:53	5.46K
ALARM: "ScaleInAlarm" in Asia Pacific (Tokyo)	AWS Notifications	25/02/26	5.46K
ALARM: "ScaleOutAlarm-CF" in Asia Pacific (Tokyo)	AWS Notifications	25/02/26	5.46K
ALARM: "ScaleInAlarm" in Asia Pacific (Tokyo)	AWS Notifications	25/02/26	5.46K
ALARM: "ScaleOutAlarm-CF" in Asia Pacific (Tokyo)	AWS Notifications	25/02/26	15.13K
ALARM: "ScaleInAlarm" in Asia Pacific (Tokyo)	AWS Notifications	25/02/26	15.15K
ALARM: "ScaleOutAlarm-CF" in Asia Pacific (Tokyo)	AWS Notifications	25/02/26	31.89K
ALARM: "ScaleInAlarm" in Asia Pacific (Tokyo)	AWS Notifications	25/02/26	15.15K

件名	送信者	操作を選択
ALARM: "ScaleOutAlarm-CF" in Asia Pacific (Tokyo)	AWS Notifications <no-reply@sns.amazonaws.com>	

```
{
    "Type" : "Notification",
    "MessageId" :
    "TopicArn" :
    "Subject" :
    "Message" :
        26708:43:54.465
        (70.0) (minimum
        1:584791980123
}
```

件名	送信者	日時	サイズ
ALARM: "ScaleInAlarm" in Asia Pacific (Tokyo)	AWS Notifications	09:09:52	5.46K
ALARM: "ScaleOutAlarm-CF" in Asia Pacific (Tokyo)	AWS Notifications	09:04:36	5.46K
ALARM: "ScaleOutAlarm-CF" in Asia Pacific (Tokyo)	AWS Notifications	08:51:36	5.46K
ALARM: "ScaleInAlarm" in Asia Pacific (Tokyo)	AWS Notifications	08:44:53	5.46K