

# AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈構築②〉）

AWS



×

ZABBIX

×



## AmazonConnectによる自動電話通知 （7.複数連絡先への電話通知〈構築②〉）

2021.11.12 2021.10.30

[【前回】 AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈構築①〉）](#)[【次回】 AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈構築③〉）](#)[【簡易版】 AmazonConnectによる自動電話通知（まとめ）](#)

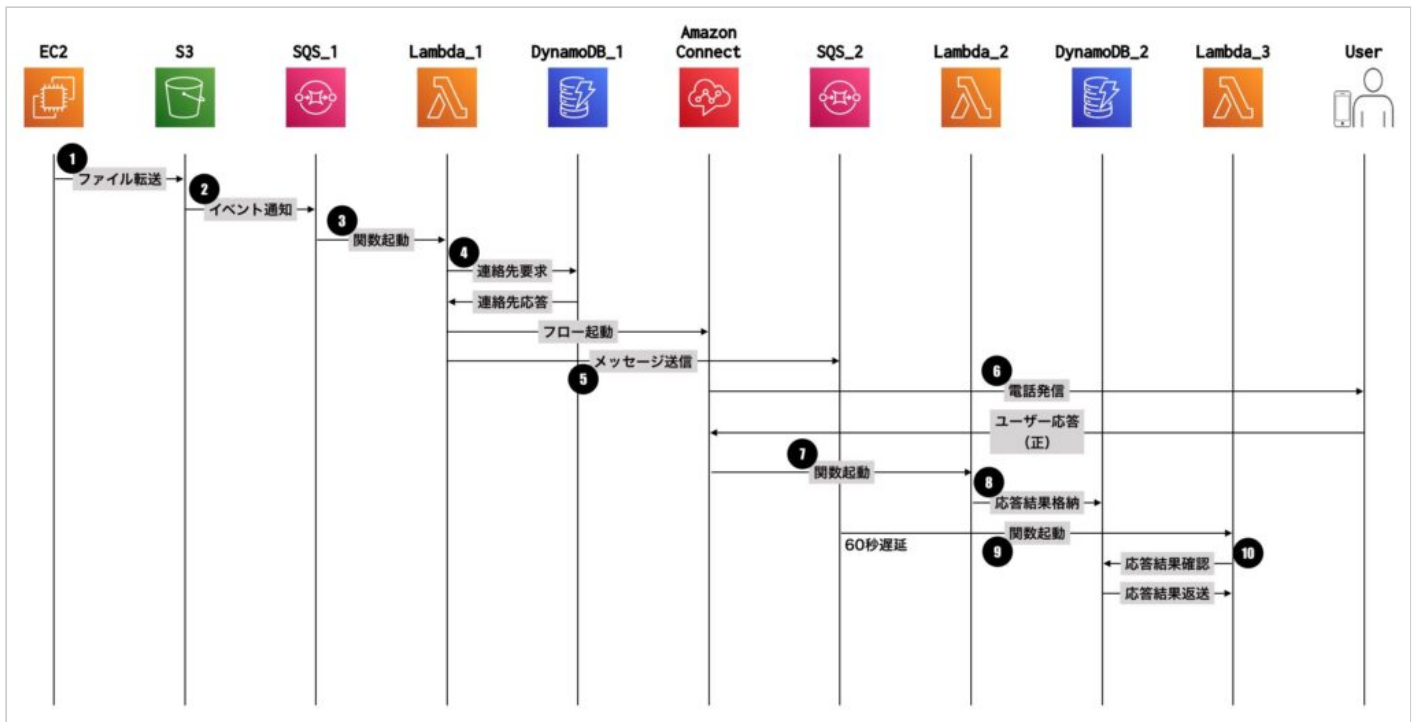
監視サーバーで障害を検知した際に、自動で電話通知できるようにしていきます。ネットワークエンジニアも利用することの多い監視サーバー(Zabbix)で障害検知し、AWS上のAmazonConnectを利用し自動電話を発信します。

今回は下記の条件を満たせるようにAWSの各サービスを利用して自動電話通知の仕組みを導入します。

- 複数の通知先を登録した連絡先リストを持たせる。
- 連絡先リストに優先度(通知順)を設定する。
- 優先度が高い人に最初に電話する。
- 応答が無かった場合、次の優先度の人に順番に電話する。
- 連絡先リストの最後まで電話しても応答が無かった場合、最初に戻って継続する。

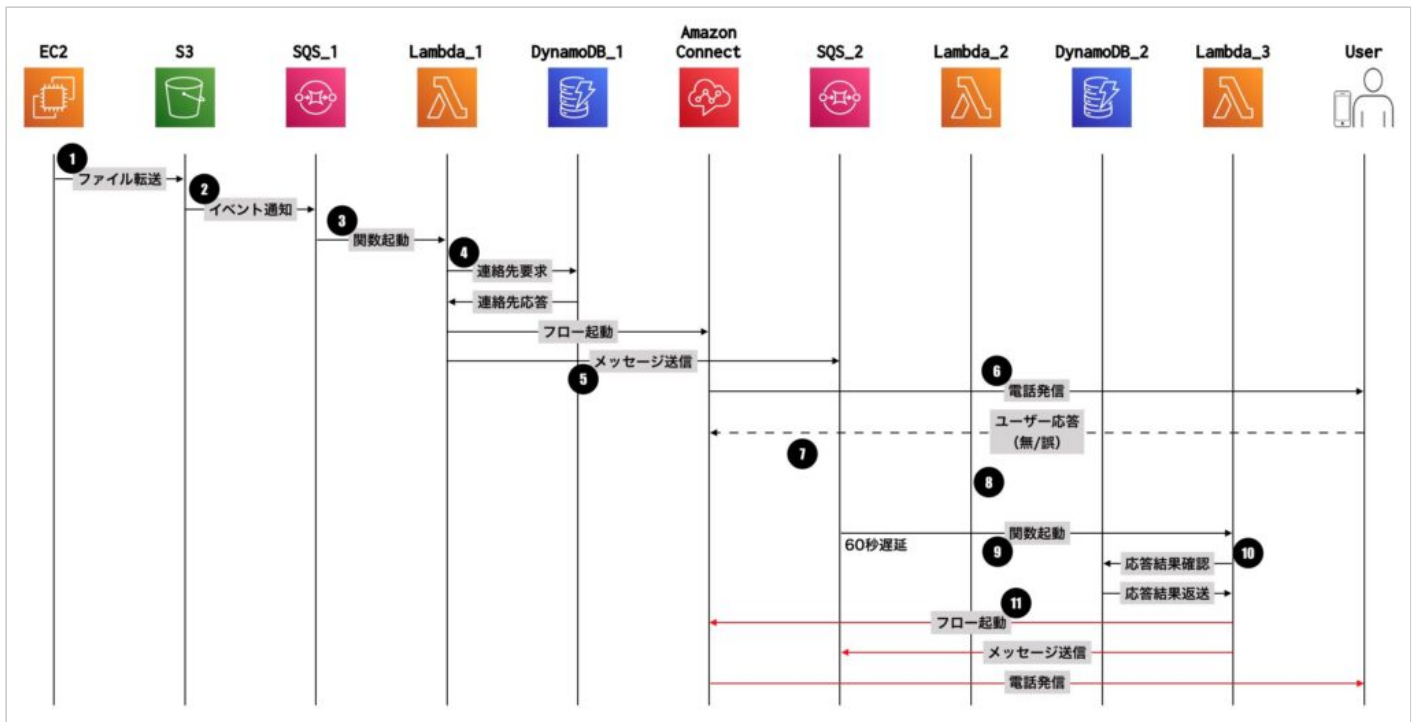
## 自動電話通知フロー

### 電話に応答した場合のフロー



1. EC2上の監視サーバーで障害を検知し、S3へトリガーファイルを格納
2. S3のイベント通知機能で、SQS\_1にメッセージを送信
3. SQS\_1をトリガーとして、Lambda\_1を起動
4. Lambda\_1がDynamoDB\_1から連絡先を取得し、AmazonConnectを起動
5. Lambda\_1がAmazonConnectを起動すると同時に、SQS\_2へメッセージを送信
6. AmazonConnectがユーザーへ自動電話通知を実施
7. ユーザーが正常応答し、AmazonConnectがLambda\_2を起動
8. Lambda\_2が応答結果をDynamoDB\_2に保存(応答OK)
9. 60秒後にSQS\_2をトリガーとしてLambda\_3を起動
10. Lambda\_3がDynamoDB\_2の応答結果を確認(正常応答しているため、何もせずに処理完了)

## 電話に応答しなかった場合のフロー



1. EC2上の監視サーバーで障害を検知し、S3へトリガーファイルを格納
2. S3のイベント通知機能で、SQS\_1にメッセージを送信
3. SQS\_1をトリガーとして、Lambda\_1を起動
4. Lambda\_1がDynamoDB\_1から連絡先を取得し、AmazonConnectを起動
5. Lambda\_1がAmazonConnectを起動すると同時に、SQS\_2へメッセージを送信
6. AmazonConnectがユーザーへ自動電話通知を実施
7. ユーザーが正常応答せず、AmazonConnectがLambda\_2を起動
8. Lambda\_2が応答結果をDynamoDB\_2に保存(応答NG)
9. 60秒後にSQS\_2をトリガーとしてLambda\_3を起動
10. Lambda\_3がDynamoDB\_2の応答結果を確認
11. 正常応答していないため、再度AmazonConnectを起動(以降、5から繰り返す)

## SQS\_1を作成(S3からイベント通知を受け取るキュー)

AWSマネジメントコンソール上で、「sqs」を検索します。



「キューを作成」をクリックします。



下記の通り選択・入力し、「キューを作成」をクリックします。

タイプ: 「標準」を選択

名前: 任意の名前を入力 ※ここでは、“amazonconnect-queue-trigger”としています。

設定: デフォルト

アクセスポリシー: 「アドバンスド」を選択し、以下を記載します。

※赤アンダーライン箇所は、入力されているまま修正不要です。

※黄アンダーライン箇所は、作成したS3バケット名を指定します。

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__owner_statement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "SQS:*",
      "Resource": "arn:aws:sqs:ap-northeast-1:XXXXXXXXXXXX:amazonconnect-queue-
trigger",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:s3:*:*:amazonconnect-alert-notification-bucket"
        }
      }
    }
  ]
}
```



Amazon SQS > キュー > キューを作成

## キューを作成

### 詳細

#### タイプ

アプリケーションまたはクラウドインフラストラクチャのキュータイプを選択します。

📘 キューの作成後にキュータイプを変更することはできません。

#### 標準 情報

少なくとも 1 回の配信、メッセージの順序は保持されません

- At-least once 配信
- ベストエフォート型の順序

#### FIFO 情報

先入れ先出し配信、メッセージの順序が保持されます

- FIFO 配信
- 1 回のみ処理

#### 名前

amazonconnect-queue-trigger

キュー名は大文字と小文字が区別され、80 文字まで入力できます。英数字、ハイフン (-)、アンダースコア (\_) を使用できます。

## 設定

最大メッセージサイズ、他のコンシューマーに対する可視性、およびメッセージの保持期間を設定します。 [情報](#)

### 可視性タイムアウト [情報](#)

30 秒 ▼

0 秒～12 時間の間である必要があります。

### メッセージ保持期間 [情報](#)

4 日 ▼

1 分～14 日の間である必要があります。

### 配信遅延 [情報](#)

0 秒 ▼

0 秒～15 分の間である必要があります。

### 最大メッセージサイズ [情報](#)

256 KB

1 KB～256 KB の間である必要があります。

### メッセージ受信待機時間 [情報](#)

0 秒

0～20 秒の間である必要があります。

## アクセスポリシー

キューにアクセスできるユーザーを定義します。 [情報](#)

### メソッドを選択

☐ ベーシック  
シンプルな基準を使用して、基本的なアクセスポリシーを定義します。

☒ アドバンスド  
JSON オブジェクトを使用して、高度なアクセスポリシーを定義します。

```
1 {
2   "Version": "2008-10-17",
3   "Id": "__default_policy_ID",
4   "Statement": [
5     {
6       "Sid": "__owner_statement",
7       "Effect": "Allow",
8       "Principal": {
9         "AWS": "*"
10      },
11      "Action": "SQS:*",
12      "Resource": "arn:aws:sqs:ap-northeast-1:██████████:amazonconnect-queue-trigger",
13      "Condition": {
14        "ArnLike": {
15          "aws:SourceArn": "arn:aws:s3:*:*:amazonconnect-alert-notification-bucket"
16        }
17      }
18    }
19  ]
20 }
21
```

[ポリシージェネレーター](#)

### ▶ 暗号化 - オプション

Amazon SQS では、デフォルトで転送時の暗号化が提供されます。キューに保管時の暗号化を追加するには、サーバー側の暗号化を有効にします。 [情報](#)

### ▶ デッドレターキュー - オプション

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。 [情報](#)

### ▶ タグ - オプション

タグは、AWS リソースに割り当てられたラベルです。タグを使用して、リソースを検索およびフィルタリングしたり、AWS コストを追跡したりできます。

[詳細はこちら](#)

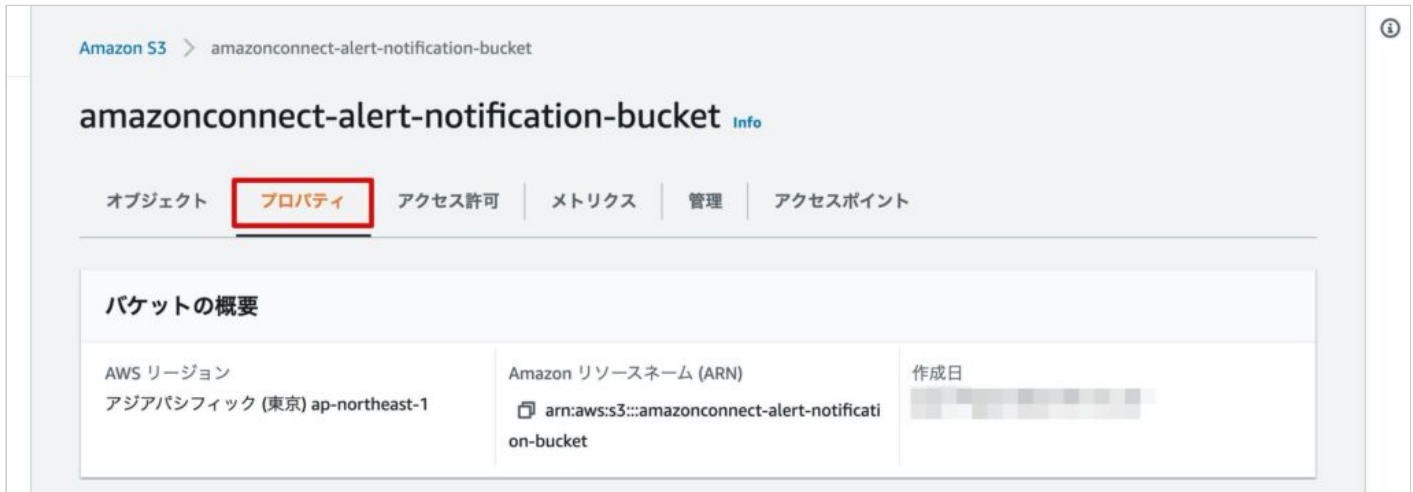
キャンセル

キューを作成



## S3のイベント通知設定

作成したS3バケットのプロパティを選択します。



「イベント通知を作成」をクリックします。



下記の通り選択・入力し、「変更の保存」をクリックします。

イベント名：任意の名前を入力 ※ここでは、“amazonconnect-launch”としています。

イベントタイプ：「すべてのオブジェクト作成イベント」にチェック

送信先：「SQSキュー」を選択

SQSキューを特定：「SQSキューから選択する」を選択

SQSキュー：作成したSQS\_1を選択

## イベント通知を作成 Info

通知設定は、Amazon S3 が発行するイベントと、Amazon S3 が通知を送信する送信先を識別します。 [詳細](#)

### 一般的な設定

#### イベント名

amazonconnect-launch

イベント名には最大 255 文字まで使用できます。

#### プレフィックス - オプション

指定された文字で始まるキーがあるオブジェクトへの通知を制限します。

images/

#### サフィックス - オプション

指定した文字で終わるキーがあるオブジェクトへの通知を制限します。

.jpg

### イベントタイプ

通知を受信するイベントのタイプを少なくとも 1 つ指定します。 [詳細](#)

#### ☒ すべてのオブジェクト作成イベント

s3:ObjectCreated:\*

##### ☒ PUT

s3:ObjectCreated:Put

##### ☒ POST

s3:ObjectCreated:Post

##### ☒ コピーする

s3:ObjectCreated:Copy

##### ☒ 完了したマルチパートアップロード

s3:ObjectCreated:CompleteMultipartUpload

#### ☐ すべてのオブジェクト削除イベント

s3:ObjectRemoved:\*

##### ☐ 完全に削除されました

s3:ObjectRemoved:Delete

##### ☐ 作成された削除マーカー

s3:ObjectRemoved:DeleteMarkerCreated

#### ☐ オブジェクトイベントを復元

##### ☐ 復元が開始されました

s3:ObjectRestore:Post

##### ☐ 復元が完了しました

s3:ObjectRestore:Completed

#### ☐ 低冗長化ストレージ (RRS) オブジェクト消失イベント

s3:ReducedRedundancyLostObject

#### ☐ レプリケーションイベント

##### ☐ レプリケーション時間制御: オブジェクトが 15 分のしきい値を超えました

s3:Replication:OperationMissedThreshold

##### ☐ レプリケーション時間制御: 15 分のしきい値経過後にレプリケートされるオブジェクト

s3:Replication:OperationReplicatedAfterThreshold



- ☐ レプリケーションによって追跡されないオブジェクト  
s3:Replication:OperationNotTracked
- ☐ オブジェクトのレプリケートに失敗しました  
s3:Replication:OperationFailedReplication

## 送信先

**i** Amazon S3 が送信先にメッセージを発行する前に、関連する API を呼び出して SNS トピック、SQS キュー、または Lambda 関数にメッセージを発行するために必要なアクセス許可を Amazon S3 の原則に付与する必要があります。 [詳細](#)

### 送信先

イベントを発行する送信先を選択します。 [詳細](#)

- ☐ Lambda 関数  
S3 イベントに基づいて Lambda 関数スクリプトを実行します。
- ☐ SNS トピック  
E メール、SMS、または HTTP エンドポイントに通知を送信します。
- ☒ SQS キュー  
サーバーによる読み込みのため SQS キューに通知を送信します。

### SQS キューを特定

- ☒ SQS キュー から選択する
- ☐ SQS キュー ARN を入力

### SQS キュー

amazonconnect-queue-rriger ▼

キャンセル

変更の保存

フィードバック 日本語 ▼

© 2008 - 2021, Amazon Web Services, Inc. または子会社

イベント通知が追加されたことを確認します。

イベント通知 (1)  
バケットで特定のイベントが発生したときに通知を送信します。 [詳細](#)

編集

削除

イベント通知を作成

<input type="checkbox"/>	名前	イベントタイプ	フィルター	送信先タイプ	送信先
<input type="checkbox"/>	amazonconnect-launch	すべてのオブジェクト作成イベント	-	SQS キュー	amazonconnect-queue-rriger <a href="#">🔗</a>

Transfer Acceleration  
高速化エンドポイントを使用してデータ転送を高速化します。 [詳細](#)

編集

Transfer Acceleration  
無効

## イベント通知設定時のエラー

下記のエラーが表示される場合は、SQSのアクセスポリシーを確認してください。

## 不明なエラー

予期しないエラーが発生しました。

Unable to validate the following destination configurations



### 不明なエラー

予期しないエラーが発生しました。

#### ▼ API レスポンス

Unable to validate the following destination configurations

キャンセル

変更の保存

## S3からSQS\_1へのイベント通知確認

SQSの「メッセージを送受信」をクリックします。

The screenshot shows the Amazon SQS console interface. The breadcrumb navigation at the top reads 'Amazon SQS > キュー > amazonconnect-queue-rriger'. The queue name 'amazonconnect-queue-rriger' is displayed prominently. To the right of the name are four buttons: '編集', '削除', 'クリア', and 'メッセージを送受信'. The 'メッセージを送受信' button is highlighted with a red rectangular border. Below the queue name, there is a '詳細 情報' section. This section contains a table with the following information:

名前	タイプ	ARN
amazonconnect-queue-rriger	標準	arn:aws:sqs:ap-northeast-1: :amazonconnect-queue-rriger
暗号化	URL	デッドレターキュー
-	https://sqs.ap-northeast-1.amazonaws.com/ /amazonconnect-queue-rriger	-

At the bottom left of the details section, there is a link that says '▶ さらに表示'.

この画面上でSQS上のメッセージを確認できます。

The screenshot shows the Amazon SQS console interface. The top navigation bar includes 'Amazon SQS', 'キュー', 'amazonconnect-queue-trigger', and 'メッセージを送受信'. The main heading is 'メッセージを送受信' with a subtitle 'キューとの間でメッセージを送受信します。'.

The 'メッセージを送信' (Send Message) section includes a 'メッセージ本文' (Message Body) input field, a '配信遅延' (Delivery Delay) dropdown set to '0' seconds, and a 'メッセージ属性 - オプション' (Message Attributes - Options) link. Buttons for 'コンテンツをクリア' (Clear Content) and 'メッセージを送信' (Send Message) are present.

The 'メッセージを受信' (Receive Message) section includes buttons for 'ポーリング設定を編集' (Edit Polling Settings), 'ポーリングを停止' (Stop Polling), and 'メッセージをポーリング' (Poll for Messages). It displays statistics: '利用可能なメッセージ' (Messages Available) at 0, 'ポーリング期間' (Polling Period) at 30, '最大メッセージ数' (Maximum Number of Messages) at 10, and 'ポーリングの進行状況' (Polling Progress) at 0% (0 messages received per second). Below this is a search bar for messages and a table with columns for 'ID', '送信済み' (Sent), 'サイズ' (Size), and '受信数' (Received). A message is not currently displayed, with a note: 'メッセージがありません。キュー内のメッセージを表示するには、メッセージをポーリングします。' (No messages. To display messages in the queue, poll for messages.) and a 'メッセージをポーリング' (Poll for Messages) button.

The footer contains 'フィードバック' (Feedback), '日本語' (Japanese), copyright information '© 2008 - 2021, Amazon Web Services, Inc. またはその関連会社。無断転用禁止。' (All rights reserved. No unauthorized reproduction.), 'プライバシーポリシー' (Privacy Policy), '利用規約' (Terms of Use), and 'Cookie の設定' (Cookie Settings).

EC2から下記のコマンドを実施し、S3へファイルをアップロードします。※黄色アンダーライン箇所は、作成したS3バケット名を指定してください。

```
touch /tmp/test.txt
aws s3 cp /tmp/test.txt s3://amazonconnect-alert-notification-bucket
```

```
[ec2-user@ip-10-0-0-100 ~]$ touch /tmp/test.txt
[ec2-user@ip-10-0-0-100 ~]$
[ec2-user@ip-10-0-0-100 ~]$ aws s3 cp /tmp/test.txt s3://amazonconnect-alert-
notification-bucket
upload: ../../tmp/test.txt to s3://amazonconnect-alert-notification-
bucket/test.txt
[ec2-user@ip-10-0-0-100 ~]$
```

「メッセージをポーリング」をクリックすると、メッセージを受信していることがわかります。

Amazon SQS > キュー > amazonconnect-queue-trigger > メッセージを送受信

## メッセージを送受信

キューとの間でメッセージを送受信します。

**メッセージを送信** 情報 コンテンツをクリア メッセージを送信

メッセージ本文  
キューに送信するメッセージを入力します。

配信遅延 情報  
 秒  
0秒～15分の間である必要があります。  
▶ **メッセージ属性 - オプション** 情報

**メッセージを受信** 情報 ポーリング設定を編集 ポーリングを停止 **メッセージをポーリング**

利用可能なメッセージ: 1    ポーリング期間: 30    最大メッセージ数: 10    ポーリングの進行状況: 0.2 件受信/秒

メッセージ (1) 詳細を表示 削除

<input type="checkbox"/>	ID	送信済み	サイズ	受信数
<input type="checkbox"/>	98b5f708-abb4-4c99-b389-f6c0f81dd9bf	<div></div>	807 バイト	1

メッセージIDをクリックすると、下記の通りS3バケットよりメッセージが送られていることがわかります。

メッセージ本文  
キューに送信するメッセージを入力します。

配信遅延 情報  
 秒  
0秒～15分の間である必要があります。  
▶ **メッセージ属性 - オプション** 情報

**メッセージを受信** 情報 ポーリング設定を編集 ポーリングを停止 **メッセージをポーリング**

利用可能なメッセージ: 1    ポーリング期間: 30    最大メッセージ数: 10    ポーリングの進行状況: 0.2 件受信/秒

メッセージ (1) 詳細を表示 削除

<input type="checkbox"/>	ID	送信済み	サイズ	受信数
<input type="checkbox"/>	98b5f708-abb4-4c99-b389-f6c0f81dd9bf	<div></div>	807 バイト	1

**メッセージ: 98b5f708-abb4-4c99-b389-f6c0f81dd9bf**  
詳細 **本文** 属性  

```
{
  "name": "amazonconnect-alert-notification-bucket",
  "owneridentity": {
    "principalId": "AQCN6S1PDKJNL",
    "arn": "arn:aws:s3:::amazonconnect-alert-notification-bucket",
    "object": {
      "key": "test.txt",
      "size": 0,
      "eTag": "d41d8cd98f00b204e9800998ecf8427e",
      "sequencer": "006173EB5A9D832F53"
    }
  }
}
```

完了

以上で、AmazonConnectによる自動電話通知（7.複数連絡先への電話通知〈構築②〉）の説明は完了です。

