

Kinesis Data Analytics を使った リアルタイム IoT デバイス モニタリング

AWS 実装ガイド

Ryan Nienhuis

Tom Horton

Rahul Sareen

2018 年 5 月

最終更新日: 2019 年 12 月 ([改訂](#)を参照)



Copyright (c) 2019 by Amazon.com, Inc. or its affiliates.

「Kinesis Data Analytics を使ったリアルタイム IoT デバイスモニタリング」ソリューションは、

<https://www.apache.org/licenses/LICENSE-2.0> で閲覧可能な Apache ライセンスバージョン 2.0 の条項に基づいてライセンスされます。

目次

概要.....	4
コスト.....	5
アーキテクチャの概要.....	5
ソリューションコンポーネント	6
Amazon Kinesis Data Analytics アプリケーション.....	6
Amazon DynamoDB.....	7
デバイスマニタリングダッシュボード.....	7
異常検出.....	8
考慮事項	9
ソリューションの更新.....	9
デプロイ可能な AWS リージョン	9
AWS CloudFormation テンプレート	9
自動デプロイ	10
スタックを起動する.....	10
セキュリティ	12
暗号化.....	12
Amazon CloudFront	12
その他のリソース.....	13
付録 A: コードコンポーネント	14
SQL クエリ	14

AWS Lambda.....	14
JavaScript.....	15
HTML 要素.....	15
付録 B: ダッシュボードのカスタマイズ	16
ステップ 1. デモスクリプトを更新して、新しいデータを送信する	16
ステップ 2. ソーススキーマにメトリクスを追加する	16
ステップ 3. アプリケーションの SQL コードを変更する	17
ステップ 4. AWS Lambda のコードを更新する	18
ステップ 5. JavaScript のコードを更新する	19
ステップ 6. ウェブサイトアセットを更新する	20
付録 C: 匿名データの収集.....	20
ソースコード	22
ドキュメントの改訂	22

このガイドについて

この実装ガイドでは、アマゾンウェブサービス (AWS) クラウドに「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションをデプロイするためのアーキテクチャ上の考慮事項と設定手順について説明します。これには、セキュリティと可用性に関する AWS のベストプラクティスを使用してこのソリューションをデプロイするために必要な AWS のサービスを起動、設定、および実行する [AWS CloudFormation](#) テンプレートへのリンクが含まれています。

このガイドは、AWS クラウドにおけるアーキテクチャの設計の実務経験がある IT インフラストラクチャアーキテクト、開発者、DevOps プロフェッショナルを対象としています。

概要

IoT デバイスをリアルタイムでモニタリングできれば、IoT デバイスの信頼性、可用性、パフォーマンスを維持するのに役立つ有益なインサイトを得ることができます。デバイスの接続とアクティビティに関する時系列データも追跡できます。このインサイトは、変化する条件や新たに発生する状況にすばやく対応するのに役立ちます。

アマゾンウェブサービス (AWS) では、強力で柔軟かつ使いやすいサービスの包括的なセットを提供しており、インサイトと実用的な情報をリアルタイムで抽出できるようにします。[Amazon Kinesis](#) では、AWS で[データをストリーミング](#)するためのプラットフォームで、あらゆる規模のストリーミングデータを費用対効果の高い方法で処理するための主要な機能を提供しています。Amazon Kinesis の機能には [Amazon Kinesis Data Analytics](#) が含まれており、新しいプログラミング言語や処理フレームワークを習得することなく、標準 SQL を使用してストリーミングデータをリアルタイムで処理できる最も簡単な方法です。

Amazon Kinesis Data Analytics をより簡単に活用していただけるように、AWS では、リアルタイムで IoT デバイスの接続性とアクティビティデータを収集、処理、分析、視覚化するために必要なサービスを自動的にプロビジョニングするリファレンス実装である「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションを提供しています。このソリューションは、メトリクスを分析して視覚化するためのフレームワークを提供するように設計されているため、基盤となるインフラストラクチャの管理ではなく、新しいメトリクスの追加に集中できます。

「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションでは、[AWS IoT](#) を使用してデバイスデータを取り込み、[Amazon Kinesis Data Firehose](#) がデータをアーカイブし、Amazon Kinesis Data Analytics がリアルタイムでメトリクスを計算して、[Amazon Simple Storage Service \(Amazon S3\)](#) と [Amazon DynamoDB](#) がメトリクスデータを永続的に保存します。このソリューションは、リアルタイムでデバイスの接続性メトリクスを視覚化するダッシュボードを備えています。

コスト

このリファレンスデプロイメントの実行中に使用した AWS サービスのコストは、お客様の負担となります。公開時点で、米国東部（バージニア北部）リージョンで、デフォルト設定でこのソリューションを実行するためのベースラインのコストは、**1 ヶ月あたり約 500 USD** です。このコスト見積もりでは、ソリューションが常時接続されている 1,000 台のデバイスマonitoringして、10 秒毎に 1 回（1 か月あたり約 250 万件のメッセージ）データを送信することを前提としています。月額費用は、接続されるデバイスの数と、それらのデバイスがデータを送信する頻度によって異なることにご注意ください。料金は変更される可能性があります。詳細については、このソリューションで使用する AWS の各サービスの料金表ウェブページを参照してください。

アーキテクチャの概要

このソリューションをデプロイすると、AWS クラウドに次の環境が構築されます。

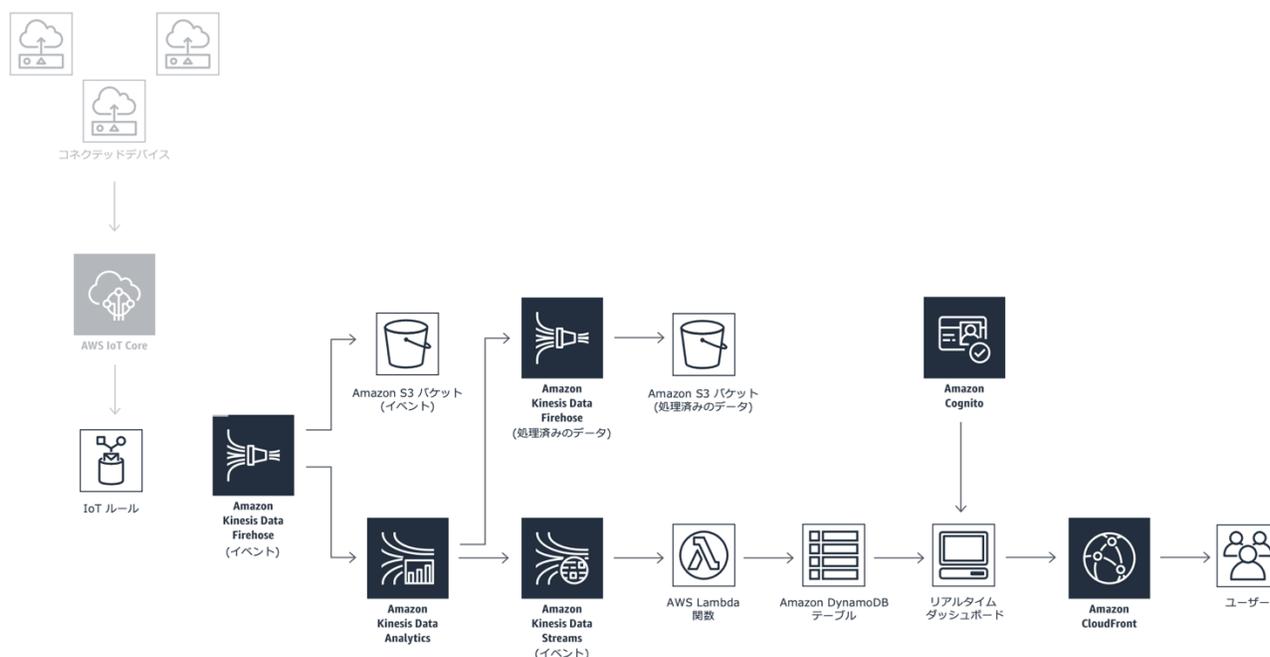


図 1: 「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマonitoring」ソリューションのアーキテクチャ

AWS CloudFormation テンプレートは、AWS IoT ルール、2 つの Amazon Kinesis Data Firehose 配信ストリーム、Amazon Simple Storage Service (Amazon S3) バケット、Amazon Kinesis Data

Analytics アプリケーション、AWS Lambda 関数、Amazon DynamoDB テーブル、Amazon Cognito ユーザープール、Amazon CloudFront ディストリビューション、Amazon DynamoDB テーブルに保存されているデバイスの接続性とアクティビティのメトリクスを安全に読み取って表示するリアルタイムのデバイスマニタリングダッシュボードをデプロイします。

AWS IoT が接続されたデバイスからデータを取り込むと、AWS IoT ルールが Amazon Kinesis Data Firehose 配信ストリームにデータを送信します。配信ストリームでは、Amazon S3 バケット内にイベントをアーカイブし、データを処理するために Amazon Kinesis Data Analytics アプリケーションに送信します。このアプリケーションは、AWS Lambda 関数に処理されたデータを送信して、Amazon DynamoDB テーブルにそのデータをリアルタイムで送信して保存します。また、このアプリケーションは、処理されたデータを 2 つ目の Amazon Kinesis Data Firehose 配信ストリームに送信して、Amazon S3 バケットにアーカイブします。

このソリューションは、Amazon Cognito ユーザープール、Amazon CloudFront ディストリビューション、Amazon S3 バケットでホストされるリアルタイムダッシュボードも作成して、Amazon DynamoDB テーブルに保存されているデバイスのアクティビティを安全に読み取って表示します。

ソリューションコンポーネント

Amazon Kinesis Data Analytics アプリケーション

このソリューションには、組み込みダッシュボードのメトリクスを計算する SQL ステートメントを備えた Amazon Kinesis Data Analytics アプリケーションが含まれています。このアプリケーションでは、Amazon Kinesis Data Firehose 配信ストリームからレコードを読み取り、SQL クエリを実行して、接続されている一意のデバイスの数、デバイスが接続された平均、最小、および最大時間、1 分間の時間枠でデバイスごとに送信されるデータポイントの最小数、最大数、および総数、接続されていない一意のデバイスの数、デバイスが切断された平均、最小、および最大時間、異常検出などの特定の IoT デバイスのメトリクスを送信し、それらを Amazon DynamoDB に保存します。詳細については、[付録 A](#) を参照してください。

Amazon DynamoDB

「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションでは、Amazon DynamoDB テーブル (AnalyticsTable) が作成されます。このテーブルには、Amazon Kinesis Data Analytics アプリケーションによって計算されるメトリクスに関する次の情報が保存されます。

- **MetricType:** 計算されたメトリクスの名前
- **EventTime:** イベントが生成された時刻
- **ConcurrencyToken:** [オブティミスティックロック](#)の更新イベントで使用されるトークン
- **Data:** JSON 形式のメトリクスデータ

デバイスマニタリングダッシュボード

このソリューションは、AmazonDynamoDB からのデータを 10 秒ごとに折れ線グラフと 1 分ごとに棒グラフに読み込むシンプルなダッシュボードを備えています。このダッシュボードでは、ユーザー認証に Amazon Cognito を利用し、Amazon Simple Storage Service (Amazon S3) バケットで[ホスト](#)されるウェブアセットを使用します。Amazon CloudFront は、このソリューションのウェブサイトバケットのコンテンツに対するアクセスを制限するために使用されます。

このダッシュボードでは、オープンソースの `chart.js` JavaScript ライブラリを使用して、HTML5 を使ってグラフを描きます。`index.html` ファイルには、ダッシュボードでグラフをレンダリングする HTML 要素が含まれています。`js` フォルダの `dash.js` ファイルには、ダッシュボードにメトリクスを入力する JavaScript が含まれています。Amazon Kinesis Data Analytics アプリケーションには、メトリクスを計算する SQL クエリが含まれています。詳細については、[付録 A](#) を参照してください。

このソリューションを正常に起動すると、ダッシュボードにログインするための手順が記載された E メールが届きます。

このダッシュボードは、追加のメトリクスを含めるようにカスタマイズすることもできます。詳細については、[付録 B](#) を参照してください。



図 2: 「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションのダッシュボード

異常検出

「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションでは、Amazon Kinesis 組み込みの[異常検出](#)を利用します。異常スコアがしきい値に達すると、そのイベントがこのソリューションの異常検出グラフに表示されます。例えば、このソリューションでは通常 18°C から 24°C の範囲の温度を記録するとします。その後、このソリューションが 37°C の温度を記録した場合は、異常を検知してグラフに表示します。

考慮事項

ソリューションの更新

「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションのバージョン 1.1.2 では、最新の Node.js ランタイムが使用されています。バージョン 1.0 では、Node.js 8.10 ランタイムを使用していました。このランタイムは、2019 年 12 月 31 日にサポートを終了しています。2019 年 1 月に、AWS Lambda は作成操作をブロックし、2019 年 2 月に AWS Lambda は更新操作をブロックしています。詳細については、AWS Lambda 開発者ガイドの[ランタイムサポートポリシー](#)を参照してください。

このソリューションに最新の機能を適用し、引き続き改善しながら使用するには、スタックを更新してください。

デプロイ可能な AWS リージョン

このソリューションでは Amazon Kinesis Data Firehose と Amazon Kinesis Data Analytics のサービスを使用しています。これらのサービスは、現在、特定の AWS リージョンでのみご利用いただけます。そのため、これらのサービスが利用可能な AWS リージョンでこのソリューションを起動する必要があります。AWS リージョン別の利用可能なサービスの最新の情報については、[AWS リージョン別の AWS 製品およびサービス一覧](#)を参照してください。

AWS CloudFormation テンプレート

このソリューションでは、AWS CloudFormation を使用して「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションのデプロイを自動化しています。このソリューションには次の AWS CloudFormation テンプレートが含まれており、デプロイ前にダウンロード可能です。

テンプレートを表示

real-time-iot-device-monitoring-with-kinesis.template: このテンプレートを使用して、ソリューションと全ての関連コンポーネントを起動します。デフォルト設定では、AWS IoT ルール、2 つの Amazon Kinesis Data Firehose 配信ストリーム、

3 つの Amazon Simple Storage Service (Amazon S3) バケット、Amazon Kinesis Data Analytics アプリケーション、AWS Lambda 関数、Amazon DynamoDB テーブル、Amazon Cognito ユーザーグループ、Amazon CloudFront ディストリビューション、リアルタイムダッシュボードがデプロイされますが、特定のニーズに基づいてテンプレートをカスタマイズすることもできます。

自動デプロイ

自動デプロイを開始する前に、このガイドで説明されているアーキテクチャ、およびその他の考慮事項をよくお読みください。このセクションの手順に従って、アカウントに「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションを設定してデプロイします。

デプロイ時間 : 約 10 分

スタックを起動する

この自動化された AWS CloudFormation テンプレートを使用して、「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションをデプロイします。スタックを起動する前に、考慮事項を確認してください。

注意: このソリューションの実行中に使用した AWS サービスのコストは、お客様の負担となります。詳細については、[コスト](#)セクションを参照してください。詳細については、このソリューションで使用する AWS の各サービスの料金表ウェブページを参照してください。

1. AWS マネジメントコンソールにサインインし、右側のボタンをクリックして、real-time-iot-device-monitoring-with-kinesis Amazon CloudFormation テンプレートを起動します。
独自にカスタマイズするために[テンプレートをダウンロード](#)することもできます。
2. テンプレートは、デフォルトで米国東部 (バージニア北部) リージョンで起動されます。別の AWS リージョンでこのソリューションを起動するには、コンソールのナビゲーションバーのリージョンセレクターを使用します。

ソリューションの
起動

注意: このソリューションでは Amazon Kinesis Data Firehose と Amazon Kinesis Data Analytics サービスを使用しています。これらのサービスは、現在、特定の AWS リージョンでのみご利用いただけます。そのため、このソリューションはこれらのサービスが利用可能な AWS リージョンで起動する必要があります。AWS リージョン別の利用可能なサービスの最新の情報については、[AWS リージョン別の AWS 製品およびサービス一覧](#)を参照してください。

3. **[テンプレートの指定]** ページで、正しいテンプレートを選択したことを確認し、**[次へ]** を選択します。
4. **[スタックの詳細を指定]** ページで、このソリューションのスタックに名前を割り当てます。
5. **[パラメータ]** で、テンプレートのパラメータを確認し、必要に応じて変更します。このソリューションでは、次のデフォルト値を使用します。

パラメータ	デフォルト	説明
User Name	<入力が必要>	リアルタイムダッシュボードにアクセスするユーザー名
User Email Address	<入力が必要>	ダッシュボードユーザーの E メールアドレス。起動後、コンソールログイン手順が記載された E メールがこのアドレスに送信されます。
IoT Topic to monitor	iot_device-analytics	デバイスがメッセージを送信する IoT トピックの名前

6. **[次へ]** を選択します。
7. **[スタックオプションの設定]** のページで、**[次へ]** を選択します。
8. **[レビュー]** ページで、設定を見直して確認します。テンプレートが IAM リソースを作成することを確認するチェックボックスを必ずオンにします。
9. **[スタックの作成]** を選択してスタックをデプロイします。

スタックのステータスは、AWS CloudFormation コンソールの **[ステータス]** 列で表示できます。約 10 分で **[CREATE_COMPLETE]** のステータスが表示されます。

このソリューションは、リアルタイムダッシュボードに参加するための招待メールを送信します。

10. Eメールの指示に従って、ダッシュボードにサインインしてください。

注意: 主要な UpdateDDBLambda AWS Lambda 関数に加えて、このソリューションには CustomResourceHelper AWS Lambda 関数も含まれています。この関数は、初期設定時かリソースの更新または削除時にのみ実行されます。

このソリューションを起動すると、AWS コンソールにこれらの AWS Lambda 関数が表示されますが、UpdateDDBLambda AWS Lambda 関数のみが定期的にアクティブになります。しかしながら、関連付けられたリソースを管理する必要があるため、CustomResourceHelper AWS Lambda 関数を削除しないでください。

セキュリティ

AWS インフラストラクチャでシステムを構築する場合、セキュリティ上の責任はお客様と AWS の間で共有されます。この責任共有モデルにより、ホストオペレーティングシステムと仮想化レイヤーからサービスが運用されているシステムの物理的なセキュリティに至るまでのコンポーネントについて、AWS が運用、管理、および制御します。そのため、お客様の運用上の負担を軽減するのに役立ちます。AWS のセキュリティの詳細については、[AWS クラウドセキュリティ](#)を参照してください。

暗号化

デフォルトでは、このソリューションで作成される Amazon Simple Storage Service (Amazon S3) バケットは暗号化されます。Amazon Kinesis Data Firehose 配信ストリームは暗号化されません。エンドツーエンドの暗号化では、このソリューションの配信ストリームへのアクセスを制限することをお勧めします。詳細については、[Amazon Kinesis Data Firehose によるアクセスの制御](#)を参照してください。

Amazon CloudFront

このソリューションでは、Amazon S3 バケットで[ホスト](#)される静的ウェブサイトをデプロイします。レイテンシーを減らし、セキュリティを向上させるために、このソリューションには、オリジンアクセスアイデンティティを持つ Amazon CloudFront ディストリビューションが含まれています。これは、このソリューションのウェブサイトの Amazon S3 バケットにあるコンテンツへのアクセスを制限する

のに役立つ特別な CloudFront ユーザーです。詳細については、[オリジンアクセスアイデンティティを使用して Amazon S3 コンテンツへのアクセスを制限する](#)を参照してください。

その他のリソース

AWS のサービス

- [Amazon Kinesis Data Firehose](#)
- [Amazon Kinesis Data Analytics](#)
- [AWS Lambda](#)
- [Amazon DynamoDB](#)
- [Amazon Simple Storage Service](#)
- [Amazon Cognito](#)
- [AWS CloudFormation](#)
- [Amazon CloudFront](#)

付録 A: コードコンポーネント

「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションでは、4 つの主要なコードコンポーネントを使用して、リアルタイムのダッシュボードにメトリクスを処理して表示します。Amazon Kinesis Data Analytics アプリケーション (KinesisAnalyticsApp) では、アプリケーション内のストリームに対して SQL クエリを実行し、その結果を出力します。AWS Lambda 関数 (UpdateDDBLambda) では、処理されたデータを Amazon DynamoDB テーブルに送信して保存します。JavaScript ファイル (dash.js) ではクエリの結果をグラフに入力し、HTML ファイル (index.html) ではグラフをダッシュボードにリアルタイムでレンダリングします。

SQL クエリ

この SQL クエリは、接続デバイスごとの最大データポイント (温度) を 1 分間隔で計算します。その結果は、メトリクスの名前 (PerDeviceMaxTemp) とその値とともに、出力アプリケーション内のストリーム (FAN_OUT_STREAM) に格納されます。

```
CREATE OR REPLACE PUMP per_device_max_pump AS
INSERT INTO FAN_OUT_STREAM
SELECT STREAM
  STEP(source_sql_stream_001."COL_time" BY INTERVAL '1' MINUTE) AS
eventTimeStamp,
  'PerDeviceMaxTemp',
  "device",
  0,
  'Maximum',
  MAX("temp") AS max_value
FROM source_sql_stream_001
GROUP BY "device", STEP(source_sql_stream_001.rowtime BY INTERVAL
'1' MINUTE), STEP(source_sql_stream_001."COL_time" BY INTERVAL '1'
MINUTE);
```

AWS Lambda

UpdateDDBLambda AWS Lambda 関数では、処理された PerDeviceMaxTemp データをリアルタイムで Amazon Kinesis Data Analytics アプリケーションから Amazon DynamoDB テーブルに送信して保存します。

```
type_operator_map = {
  'ConnectedDevicesCount' : max,
  'PerDeviceMaxTemp' : max,
  'PerDeviceMinTemp': min,
  'PerDeviceAvgTemp': avg,
  'DeviceTempAnomalyScore': max,
  'AvgTempValue': avg,
  'MinTempValue': min,
  'MaxTempValue': max,
  'MaxDisconnTime': max,
  'MinDisconnTime': min,
  'AvgDisconnTime': avg,
  'MaxConnTime': max,
  'MinConnTime': min,
  'AvgConnTime': avg
}
```

JavaScript

JavaScript (dash.js ファイル内) では、接続されたデバイスごとの最大データポイント (温度) をグラフに入力します。

```
var PerDeviceMaxTempParams = retrieveParams("PerDeviceMaxTemp",
maxTempPerDeviceQueryTime);

docClient.query(PerDeviceMaxTempParams, function(err, data) {
  if (err) console.log(err);
  else {
    maxTempPerDeviceQueryTime = updateHorizontalBarChart(data,
20, maxTempPerDeviceChart, maxTempPerDeviceQueryTime, splitFunc);
  }
});
```

HTML 要素

HTML 要素 (index.html ファイル内) では、SQL クエリの結果でデバイスごとの最高温度をレンダリングします。

```
<div class="row aws-mb-1">

  <div class="col-md-5 col-md-offset-1 col-xs-12">

    <div class="x_title">
      <h3>Min Temperature per Device</h3>
```

```
</div>
<div class="x_content">
  <canvas id="minTempCanvas"></canvas>
</div>
</div>
<div class="col-md-5 col-xs-12">
  <div class="x_title">
    <h3>Max Temperature per Device</h3>
  </div>
  <div class="x_content">
    <canvas id="maxTempCanvas"></canvas>
  </div>
</div>
</div>
</div>
```

付録 B: ダッシュボードのカスタマイズ

「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションのダッシュボードにはメトリクスのデフォルトセットが表示されますが、IoT デバイスからのメトリクスを含めるようにダッシュボードをカスタマイズできます。このセクションの手順に従って、最大気圧を計算するメトリクスを追加します。

ステップ 1. デモスクリプトを更新して、新しいデータを送信する

send-messages.sh スクリプトで、JSON のペイロードを更新して、圧力値を整数として送信します。この演習では、スクリプトに太字の JSON を追加します。

```
aws iot-data publish --topic "$TOPIC" --payload
"{\"id\": \"1\", \"device\": \"$DEVICE\", \"flow\": $FLOW, \"temp\": $TEMP,
  \"humidity\": $HUMIDITY, \"sound\": $SOUND, \"pressure\": 100}" --
profile "$PROFILE" --region "$REGION"
```

ステップ 2. ソーススキーマにメトリクスを追加する

この手順を使用して、ソーススキーマを新しいメトリクスで更新します。スキーマエディタの使用については、Amazon Kinesis Data Analytics 開発者ガイドの[スキーマエディタの使用](#)を参照してください。

注意: すでにカスタムメトリクスが Amazon Kinesis Data Analytics アプリケーションのアプリケーション内の入力ストリームに追加されている場合は、[ステップ 3](#) に進んでください。

1. AWS マネジメントコンソールにサインインし、Amazon Kinesis Data Analytics コンソールを開きます。
2. リストから **KinesisAnalyticsApp** アプリケーションを選択します。
3. **Real-Time Analytics** で、**[Go to SQL results]** を選択します。
4. **Source data** タブで、**[Actions]** を選択します。
5. ドロップダウンメニューで、**[Edit schema]** を選択します。
6. **[+Add column]** を選択し、次のように入力します。
 - **Column name:** `pressure`
 - **Column type:** `integer`
 - **Row path:** `$.pressure`
7. **[Save schema and update stream samples]** を選択します。
8. メトリクスが正しく追加されたことを確認するには、**[Go to SQL results]** を選択して、**Source data** タブに新しい列 (`pressure`) と該当する値が表示されていることを確認します。

ステップ 3. アプリケーションの SQL コードを変更する

この手順を使用して、新しい SQL ステートメントでアプリケーションのコードを更新します。SQL エディタの使用については、Amazon Kinesis Data Analytics 開発者ガイドの [SQL エディタの使用](#) を参照してください。

1. Amazon Kinesis Data Analytics アプリケーションの **SQL Editor** ページで、**Real-time analytics** タブを選択します。
2. 太字の SQL ステートメントを追加します。

```
CREATE OR REPLACE PUMP connected_device_pump AS INSERT INTO
FAN_OUT_STREAM
```

```
SELECT current_timestamp as eventTimeStamp, 'ConnectedDevicesCount',
'None', 0, 'Count', * FROM (
  SELECT STREAM * FROM TABLE(COUNT_DISTINCT_ITEMS_TUMBLING(
    CURSOR(SELECT STREAM * FROM source_sql_stream_001),
    'device',
    60
  )
)
);

CREATE OR REPLACE PUMP per_device_max_pump AS INSERT INTO
FAN_OUT_STREAM
SELECT STREAM
STEP(source_sql_stream_001."COL_time" BY INTERVAL '1' MINUTE) AS
eventTimeStamp,
'PerDeviceMaxPressure',
"device",
0,
'Maximum',
MAX("pressure") AS max_value
FROM source_sql_stream_001
GROUP BY "device", STEP(source_sql_stream_001.rowtime BY INTERVAL
'1' MINUTE), STEP(source_sql_stream_001."COL_time" BY INTERVAL '1'
MINUTE);
```

この SQL ステートメントでは、新しいメトリクス (PerDeviceMaxPressure) を作成して、Amazon DynamoDB テーブルに最大気圧の読み取り値を保存します。

3. [Save and run SQL] を選択します。

ステップ 4. AWS Lambda のコードを更新する

太字のコードを UpdateDDBLambda AWS Lambda 関数に追加します。

```
type_operator_map = {
  'ConnectedDevicesCount' : max,
  'PerDeviceMaxTemp' : max,
  'PerDeviceMinTemp' : min,
  'PerDeviceAvgTemp' : avg,
  'PerDeviceMaxPressure' : max,
  'DeviceTempAnomalyScore' : max,
  'AvgTempValue' : avg,
  'MinTempValue' : min,
  'MaxTempValue' : max,
  'MaxDisconnTime' : max,
  'MinDisconnTime' : min,
  'AvgDisconnTime' : avg,
  'MaxConnTime' : max,
```

```
'MinConnTime': min,  
'AvgConnTime': avg  
}
```

ステップ 5. JavaScript のコードを更新する

このソリューションでは、js フォルダを使用して Amazon S3 バケット (websitebucket) を作成します。これにはグラフにメトリクスを設定する JavaScript コードを持つ dash.js ファイルが含まれています。新しいグラフにメトリクスを設定するには、dash.js ファイルをダウンロードして、次の手順に従い、JavaScript を変更します。

1. 変数とパラメータを宣言します。この演習では、dash.js ファイルの 267 行目に太字の JavaScript を追加します。

```
var maxTempPerDeviceQueryTime = new Date(currentTime.getTime() -  
600000).toISOString().replace('T', ' ').replace('Z', '');  
  
var maxTempPerDeviceChart =  
generateHorizontalBarChart("maxTempCanvas", "Max Temp per device");  
  
var maxPressurePerDeviceQueryTime = new Date(currentTime.getTime() -  
600000).toISOString().replace('T', ' ').replace('Z', '');  
var maxPressurePerDeviceChart =  
generateHorizontalBarChart("maxPressureCanvas", "Max Pressure per  
device");
```

2. getLatestRecord 関数を変更します。この関数に太字の JavaScript を追加します。

```
var PerDeviceMinTempParams = retrieveParams("PerDeviceMinTemp",  
minTempPerDeviceQueryTime);  
var PerDeviceMaxPressureParams =  
retrieveParams("PerDeviceMaxPressure",  
maxPressurePerDeviceQueryTime);  
var AvgTempParams = retrieveParams("AvgTempValue",  
AvgTempValueQueryTime);  
  
...  
  
docClient.query(PerDeviceMaxTempParams, function(err, data) {  
    if (err) console.log(err);  
    else {  
        maxTempPerDeviceQueryTime = updateHorizontalBarChart(data,  
20, maxTempPerDeviceChart, maxTempPerDeviceQueryTime, splitFunc);  
    }  
});
```

```
docClient.query(PerDeviceMaxPressureParams, function(err, data) {
  if (err) console.log(err);
  else {
    maxPressurePerDeviceQueryTime =
    updateHorizontalBarChart(data, 20, maxPressurePerDeviceChart,
    maxPressurePerDeviceQueryTime, splitFunc);
  }
});
```

3. 変更した `dash.js` ファイルをこのソリューションの Amazon S3 バケットにアップロードします。

ステップ 6. ウェブサイトアセットを更新する

JavaScript を使用した Amazon S3 バケットには、すべての HTML 要素を含むファイル (`index.html`) があり、ダッシュボードでグラフをレンダリングします。新しいグラフを追加するには、`index.html` ファイルをダウンロードして、HTML を変更し、変更したファイルを Amazon S3 バケットにアップロードします。既存グラフの行を新しい行に置き換えるか、新しい行をファイルの末尾に追加することができます。

この演習では、`index.html` ファイルの 287 行目に次の HTML 要素を追加します。

```
<div class="row aws-mb-1">
  <div class="col-md-5 col-md-offset-1 col-xs-12">
    <div class="x_title">
      <h3>Max Pressure per Device</h3>
    </div>
    <div class="x_content">
      <canvas id="maxPressureCanvas"></canvas>
    </div>
  </div>
</div>
```

変更した `index.html` ファイルを Amazon S3 バケットにアップロードしたら、ブラウザでダッシュボードを開き、新しいグラフにメトリクスが表示されることを確認します。

付録 C: 匿名データの収集

このソリューションには、匿名の使用状況データを AWS に送信するオプションが含まれています。当社はこのデータを使用して、お客様によるこのソリューションの使用状況をよりよく理解し、提供する

サービスや製品の改善に役立っています。有効にすると、ダッシュボードが閲覧されるたびに次の情報が収集され、AWS に送信されます。

- **Solution ID:** AWS ソリューション識別子
- **Unique ID (UUID):** ソリューションのデプロイごとにランダムに生成された一意の識別子
- **Timestamp:** データ収集タイムスタンプ
- **Dashboard Views:** ダッシュボードが閲覧される回数
- **Processed Metrics:** メトリクスの更新回数

このオプションで収集されたデータは AWS に帰属します。データ収集には、[AWS プライバシーポリシー](#)が適用されます。この機能を無効にするには、次のいずれかのタスクを実行します。

a) AWS CloudFormation テンプレートマッピングセクションを次のように変更します。

```
Solution:
  Data:
    SendAnonymousUsageData: "True"
```

を次に変更します。

```
Solution:
  Data:
    SendAnonymousUsageData: "False"
```

または

b) このソリューションを起動した後で、AWS Lambda コンソールで CustomResourceHelper を見つけ、**SEND_ANONYMOUS_DATA** 環境変数を False に設定します。

ソースコード

[GitHub リポジトリ](#)にアクセスして、このソリューションのテンプレートとスクリプトをダウンロードし、カスタマイズした上で他のユーザーと共有できます。

ドキュメントの改訂

日付	変更
2018 年 5 月	初回リリース
2018 年 12 月	Amazon S3 パケットでホストする静的ウェブサイトの Amazon CloudFront ディストリビューションに関する情報を追加
2019 年 8 月	このソリューションの AWS Lambda 関数を最新の Node.js ランタイムにアップグレード
2019 年 12 月	Node.js の更新のサポートに関する情報を追加

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

注意

本ドキュメントは情報提供のみを目的としています。本書の発行時点における AWS の現行製品と慣行を表したものであり、それらは予告なく変更されることがあります。お客様は本書の情報、および AWS 製品またはサービスの利用について、独自の評価に基づき判断する責任を負います。いずれの AWS 製品またはサービスも、明示または黙示を問わずいかなる保証も伴うことなく、「現状のまま」提供されます。本書のいかなる内容も、AWS、その関係者、サプライヤー、またはライセンサーからの保証、表明、契約的責任、条件や確約を意味するものではありません。お客様に対する AWS の責任は、AWS 契約により規定されます。本書は、AWS とお客様の間で行われるいかなる契約の一部でもなく、そのような契約の内容を変更するものでもありません。

「Kinesis Data Analytics を使ったリアルタイム IoT デバイスマニタリング」ソリューションは、<https://www.apache.org/licenses/LICENSE-2.0> で閲覧可能な Apache ライセンスバージョン 2.0 の条項に基づいてライセンスされます。