

# IoTで実現する簡単な製品異常の送信とメール通知

本Webレシピでは、お客さまの製品を模したIoTデバイスから、IoTプラットフォーム「Things Cloud」に製品の異常発生情報（アラーム）を送信し、受信したアラームをウェブ画面で可視化する方法についてご紹介します。また、アラームを受信した際に、Things Cloudの機能により管理者へのメール通知を簡単に実現する方法についてもご説明します。

難易度：★☆☆☆☆

レシピ公開日：2024年11月

## このレシピの内容

### 1. 概要

- 1-1. 本レシピで実現できること
- 1-2. 所要時間
- 1-3. 所要費用
- 1-4. 使用するサービス
- 1-5. 準備する物

### 2. 作業の流れ

- 2-1. ハードウェアの構築手順
  - Raspberry PiのセットアップとNode-RED導入
  - デバイスからアラームを送信するための設定
  - アラーム一覧の見方
  - スマートルールを使用したメール通知の設定
  - 【応用編】独自のアラーム内容を送信

### 3. 最後に

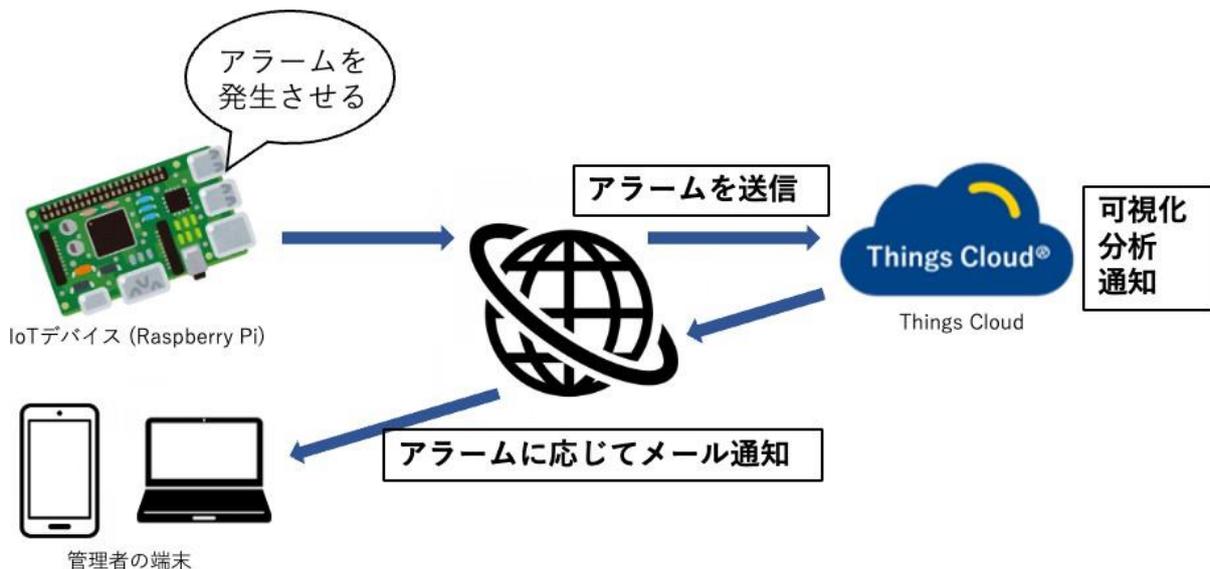
- 3-1. 次のステップ
- 3-2. Things Cloudの詳細情報とお申し込みについて

# 1.概要

IoTは、日常生活からビジネスの現場まで幅広く活用されています。特に、産業機械やプラント設備の状態監視、建機やトラクターといった産業用車両の追跡監視など、「製品の遠隔監視」は有望な活用例です。IoTを用いた遠隔監視では、遠隔地や危険区域からもリアルタイムにデータを取得できるため、コスト削減、安全性向上、迅速な対応が可能となります。

## 1-1.本レシピで実現できること

本レシピでは、Raspberry PiをIoTデバイスに見立て、アラームをThings Cloudに送信します。さらに、Things Cloud側で受信したあと、可視化を行い、さらにThings Cloudの機能を利用して以下の簡単なシステムを構築してみます。

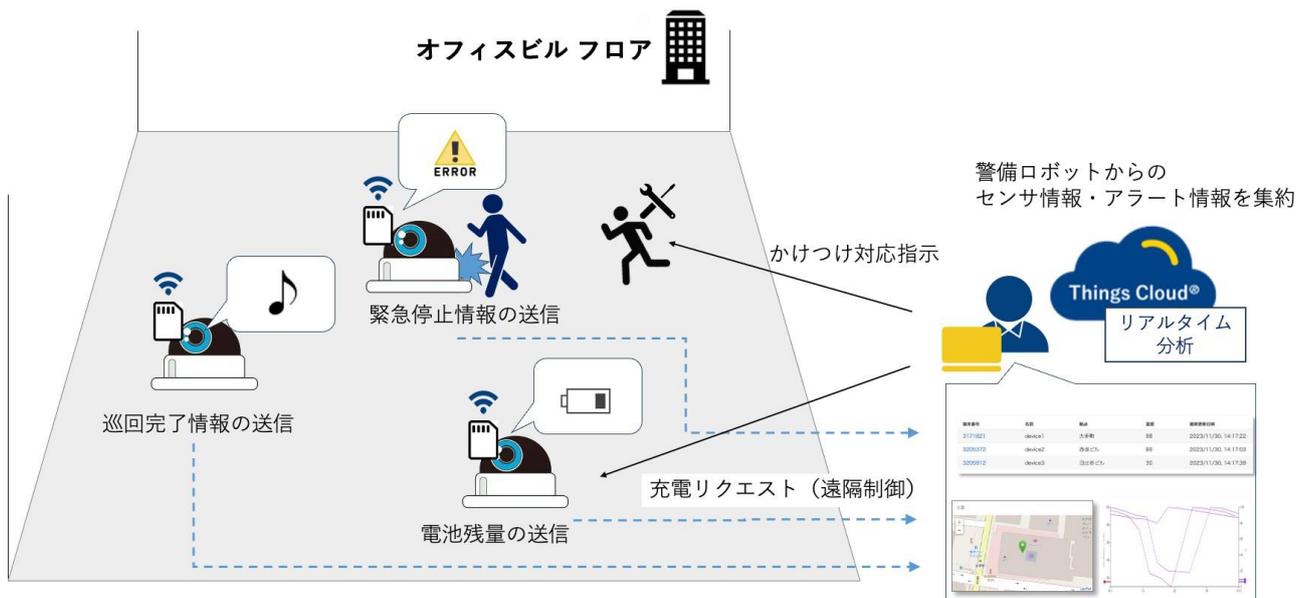


今回はIoTデバイスとしてRaspberry Piを使用しますが、お持ちのIoTデバイスでもThings Cloudにデータを送ることができればThings Cloudのアラーム設定を行うことが可能ですので、お手元のIoTデバイスのアラームでも本手順を応用することで簡単にアラームを管理することができるようになります。さまざまなIoTデバイスをThings Cloudと接続し、アラームを一元管理することにより、複数のIoT製品の遠隔監視や保守などの負担を軽減し、より効率的な運用を実現することが可能となり、コストの削減などに貢献します。

## アラーム機能の活用例

アラーム機能は、前述したとおり遠隔監視に用いられるケースが多いですが、具体的な活用例を1つご紹介します。昨今の商業施設においては、AIや5Gなどの新しい技術を用いて、警備員の代わりにロボットを導入し、ロボットが巡回業務を行うことで安心・安全のみならず、大幅な業務の効率化を実現しているケースがあります。Things Cloudで提供している機能により、このようなロボットの遠隔監視を支援し、ロボットが発生させたアラームを一元管理することが可能です。

簡単な図を以下に示します。



巡回しているロボットは、センサーデータを取得し、巡回の状況をはじめ、バッテリーの情報や緊急停止アラームなどのさまざまな情報をThings Cloudに送信します。このアラームの情報について、本レシピで紹介する機能を応用することで、可視化することはもちろん、他にも発生したアラームに対しメールで管理者に通知を行うなど、運用する上で必要となるさまざまな機能を有しています。例えば電池残量が少なくなっていることを確認した際には、Things Cloud側から遠隔制御で充電のリクエストを行ったり、さまざまなアラームを確認することで、ロボットの状況を逐一把握し、適切な対応を行うことが可能です。

このソリューションでは、一例として以下のような画面を作成し、ロボットの状態をリアルタイムに把握できるようにします。1つの画面で見たい情報を一覧で表示することができるため、運用者はロボットの状況を一目で把握することが可能です。



## 1-2.所要時間

約120分（デバイスセットアップ50分を含む）

## 1-3.所要費用

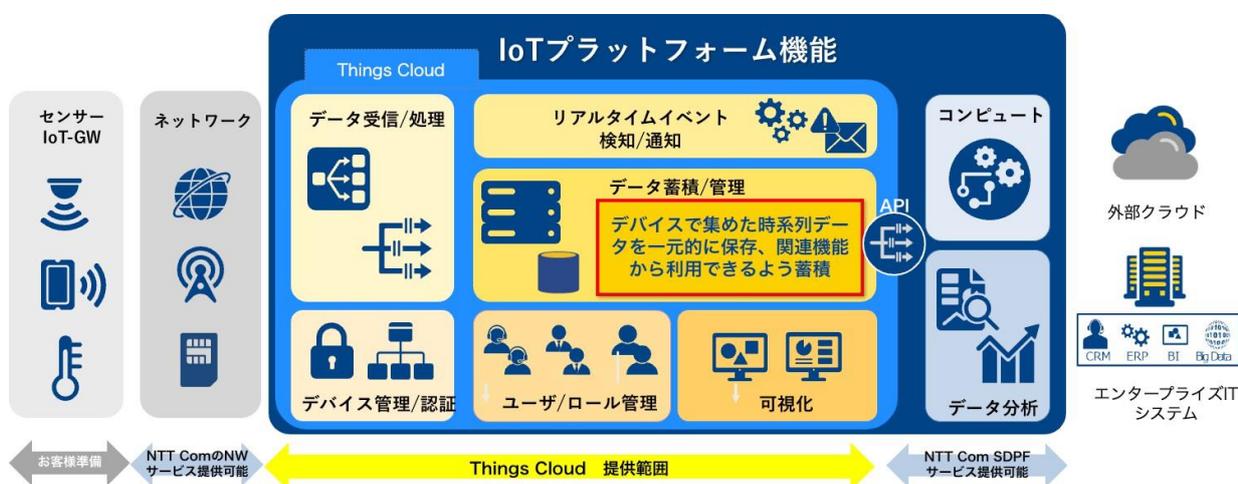
約13,500円（デバイス購入の初期費用 + サービス利用料金）

## 1-4.使用するサービス

[Things Cloud®](#)

### Things Cloudとは

多様なセンサーやデバイス接続からのデータ収集・可視化・分析、デバイスの管理などIoTに必要な機能をパッケージ化したIoTプラットフォームサービスです。IoTシステムに必要な機能の9割が組み込まれており、IoTを簡単に導入するのに適しています。また、直感的にUIをカスタマイズ可能で、用途に合った可視化画面を簡単に作成でき、さらにAPIファースト設計で柔軟なシステム連携も実現しており、幅広い使い方が可能なサービスとなっています。



## 1-5.準備する物

### ハードウェア

物品名	数量	価格	備考
Raspberry Pi (3以降のModel B)	1	約 10,000円	今回はRaspberry Pi 4 Model Bを使用
microSDカード	1	約 1,000円	16GB以上のものを推奨
microSDカードリーダー	1	約 500円	
Raspberry Pi 用電源アダプター	1	約 1,000円	5V 2A以上のものを推奨
PC	1	-	Windows PC もしくは Macを推奨

※ 金額はレシピ作成の参考価格となります。



### その他準備するもの

必要なもの	費用	備考
Things Cloudのテナント	製品IoT化パックをご契約頂くことで、Things Cloudを利用することが可能です。利用料金の詳細は <a href="#">こちら</a> をご参照ください。	-

## 2. 作業の流れ

### 2-1. ハードウェア構築手順

#### Raspberry PiのセットアップとNode-RED導入

Raspberry PiをIoTデバイスとして利用するために、OSのインストールとNode-REDの導入を行います。以下のレシピを参考に、Raspberry Piのセットアップとデバイス認証情報の作成を行ってください。

- [Raspberry Piとノーコードツール「Node-RED」を利用したIoTデバイスのセットアップ](#)

既にセットアップ済みのRaspberry Piがある場合は、Node-REDのインストールとデバイス認証情報の作成のみを行ってください。

#### デバイスからアラームを送信するための設定

ここからは、Raspberry Piに導入したNode-REDを使ってデバイスからThings Cloudにアラームを送信するための事前設定を行います。

#### Node-REDのサンプルフロー読み込み

セットアップ済みのRaspberry Piを起動し、インストールしたNode-REDの画面を開きます。自身のPCのブラウザから以下のURLにアクセスして、Node-REDにログインしてください。

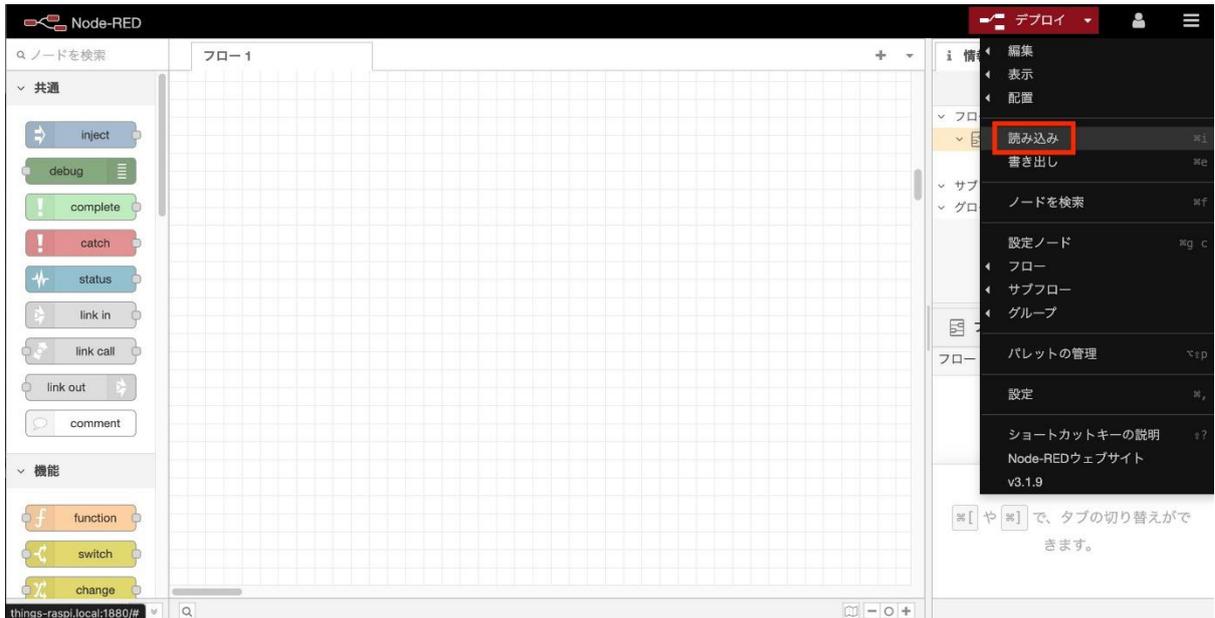
`http://things-raspi.local:1880`

- Raspberry PiとPCが同じネットワークに接続されていることを確認してください。
- `things-raspi`の部分は、自身のRaspberry Piのホスト名に置き換えてください。
- ログイン画面が表示されたら、Node-REDインストール時に指定したユーザー名とパスワードを入力してログインします。

[こちら](#)からサンプルフローをダウンロードし、Node-REDにインポートします。

## インポートの手順：

1. Node-REDの画面右上のメニューから「読み込み」を選択

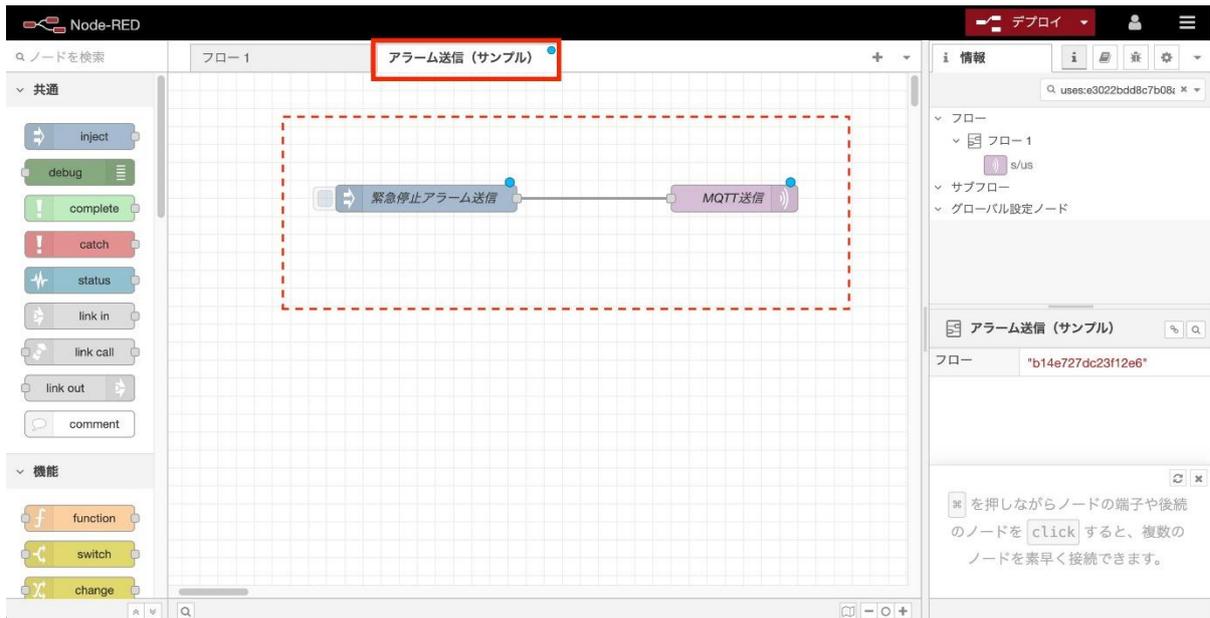


2. 「読み込むファイルを選択」からダウンロードしたファイルを選択し、「読み込み」をクリックする



3. 「アラーム送信（サンプル）」というフローが追加されるので、追加されたフロー(上部タブをクリック)に切り替えます。

以下のような画面（フロー）が表示されれば成功です。



## MQTT接続設定

デバイスからThings Cloudにデータを送信するために、今回はMQTTを利用します。以下の手順でNode-RED上のMQTTの設定を変更していきます。

### MQTTとは

MQTT (Message Queuing Telemetry Transport) は、IoTデバイス間でメッセージをやり取りするためのプロトコルです。軽量で通信速度が速く、低消費電力で動作するため、IoTデバイス間の通信に適しています。

Things CloudにおけるMQTT接続については、[こちら](#)をご参照ください。また、Things Cloudは、より一般的なHTTPを利用したREST API ([API仕様書](#)) にも対応しています。

1. 自身のPCのブラウザから以下のURLにアクセスしてNode-REDにログインし、フローエディタ画面を開きます。

`http://things-raspi.local:1880`

- Raspberry PiとPCが同じネットワークに接続されていることを確認してください。
- `things-raspi` の部分は、Raspberry Piのセットアップ時に設定したホスト名に置き換えてください。
- ログイン時のユーザー名とパスワードは、Node-REDのインストール時に設定したものを入力してください。

2. フロー画面で、「MQTT送信ノード」をダブルクリックして設定画面を開きます。次に、「サーバー」の右側にある鉛筆アイコンをクリックして、MQTTサーバーの設定を行います。



3. MQTTサーバーの設定画面が表示されたら、以下の設定を変更します。

※ その他の項目は変更不要です。

### 接続タブ

項目	説明	備考
サーバー	[テナントのドメイン]/mqtt	ログイン時のURLのドメイン部分を入力 例: test-tenant.je1.thingscloud.ntt.com/mqtt <b>注: "https://"の部分は不要</b>
ポート	8883	
TLSを使用	チェックを入れる	
クライアント	デバイスを一意に示す任意のID	独自のデバイス固有のIDを設定可能 (シリアル番号、社内の管理番号など) 例: Raspberry_Pi001

## セキュリティタブ

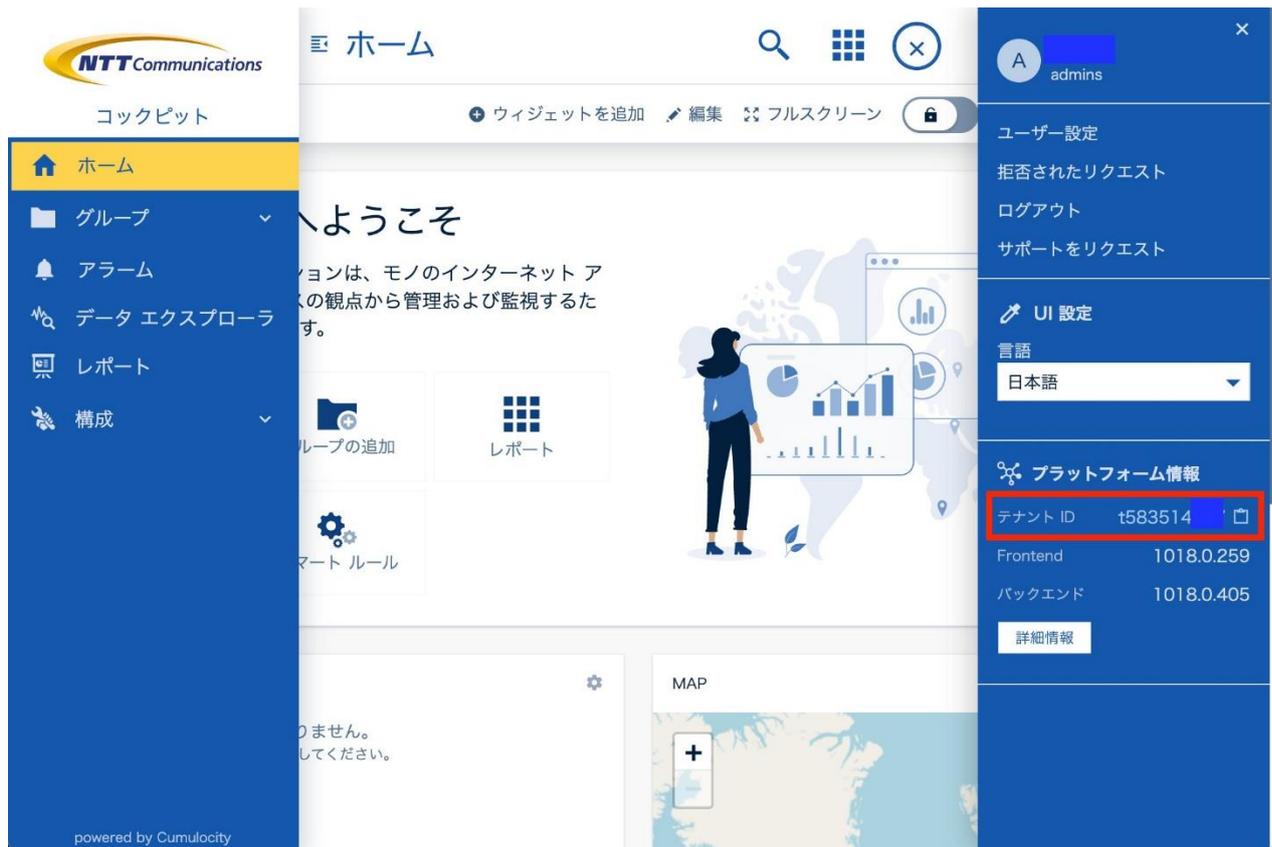
項目	説明	備考
ユーザー名	[テナントID]/[デバイスユーザー名]	テナントIDと先ほど作成したデバイスユーザー名を入力例： t12345678/username
パスワード	[デバイスユーザーのパスワード]	

The screenshot displays the configuration page for an MQTT broker node in Things Cloud. The page is titled "mqtt out ノードを編集 > mqtt-broker ノードを編集". It features a navigation bar with "削除", "中止", and "更新" buttons. The main content area is divided into tabs: "接続", "セキュリティ", and "メッセージ". The "セキュリティ" tab is currently selected. Under this tab, there are input fields for "ユーザー名" (Username) containing "t12345678/device\_raspberrypi" and "パスワード" (Password) containing ".....". Other visible fields include "名前" (Name) as "Things Cloud MQTT接続 (サンプル)", "サーバ" (Server) as "your-domain.je1.thingscloud.ntt.com/mqtt", "ポート" (Port) as "8883", "プロトコル" (Protocol) as "MQTT V3.1.1", and "クライアント" (Client) as "Device\_Name". There are also checkboxes for "自動接続" (Auto-connect), "TLSを使用" (Use TLS), and "セッションの初期化" (Initialize session).

### テナントIDの確認方法

Things CloudのテナントIDは、ログイン後に右上のユーザーアイコンをクリックした時に表示されるプラットフォーム情報の中に記載されています。一般的に"t"から始まる英数字の文字列です。

※ テナントのサブドメイン部分とは異なることに注意してください。



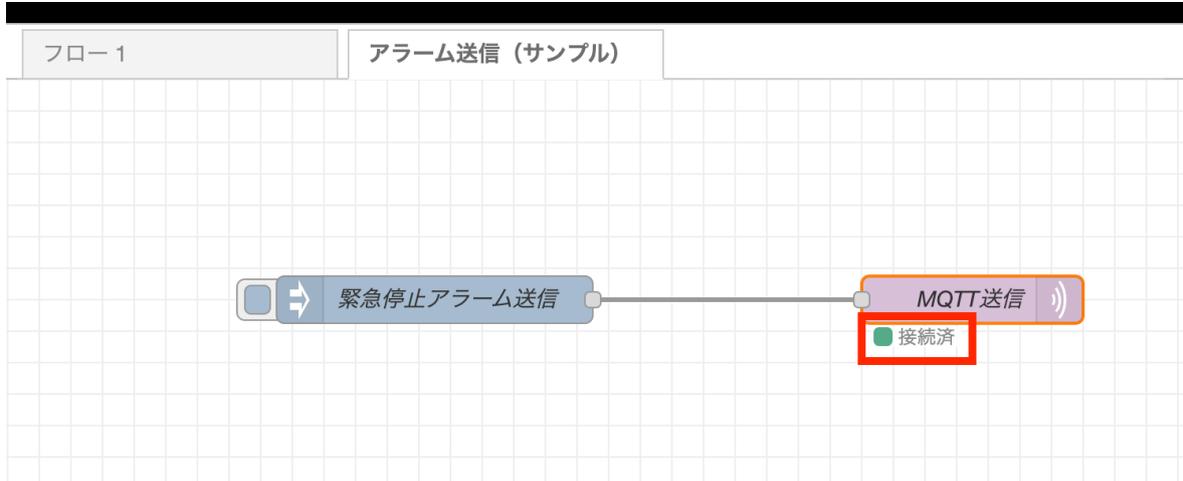
#### デバイスユーザー（デバイス認証情報）について

Things Cloudでは、デバイスが利用するための専用ユーザー（=デバイスユーザー）の利用を推奨しています。デバイスユーザーをまだ作成していない場合

は、[Raspberry Piのセットアップ手順](#)の「デバイス用の認証情報を作成」セクションを参照してください。

4. 設定が完了したら、画面右上の「更新」をクリックして設定を保存します。そのままmqtt outノードの編集画面も「完了」をクリックして閉じます。

5. 最後に、フロー画面右上の「デプロイ」をクリックして設定を反映させます。デプロイが完了したら、「MQTT送信」ノード部分が接続済（緑色）になっていれば成功です。
- ※ 接続済になっていない場合は、ノードをクリックして設定を確認してください。

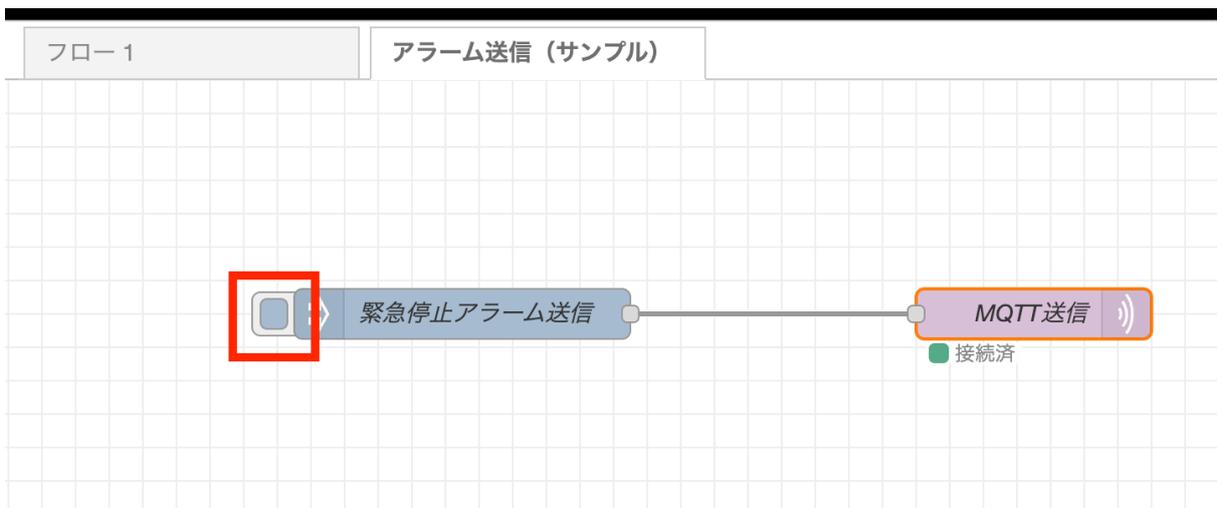


以上で、デバイスからThings Cloudにアラームを送信するための設定は完了です。

### アラーム送信の実行方法

サンプルフローには、既にアラームを送信するためのノードが設定されています。例としてデバイスの緊急停止を示すアラームが設定されています。「緊急停止アラーム送信」ノードの左にあるボタンをクリックすることで、アラームを送信することが可能です。Things Cloudにデバイスを登録するために、1度アラーム送信ボタンを押してみましょう。

デバイスからMQTTでデータを送信することで、Things Cloudのデバイス一覧に本デバイスが追加されます。次の手順で必要なため、必ず1回以上アラームを送信してください。



## アラーム一覧の見方

デバイスからThings Cloudに向けて送信したアラームは、すぐにダッシュボードで可視化し、どのようなアラームが発生したか確認することができます。アラームを可視化する設定を行い、実際にアラームを表示させる方法について紹介します。

## コックピットアプリへのログイン

ご自身のThings Cloudのテナントにアクセスし、ユーザー名とパスワードを入力してログインを行います。

NTT Communications

ログイン

ユーザー名

admin

パスワード

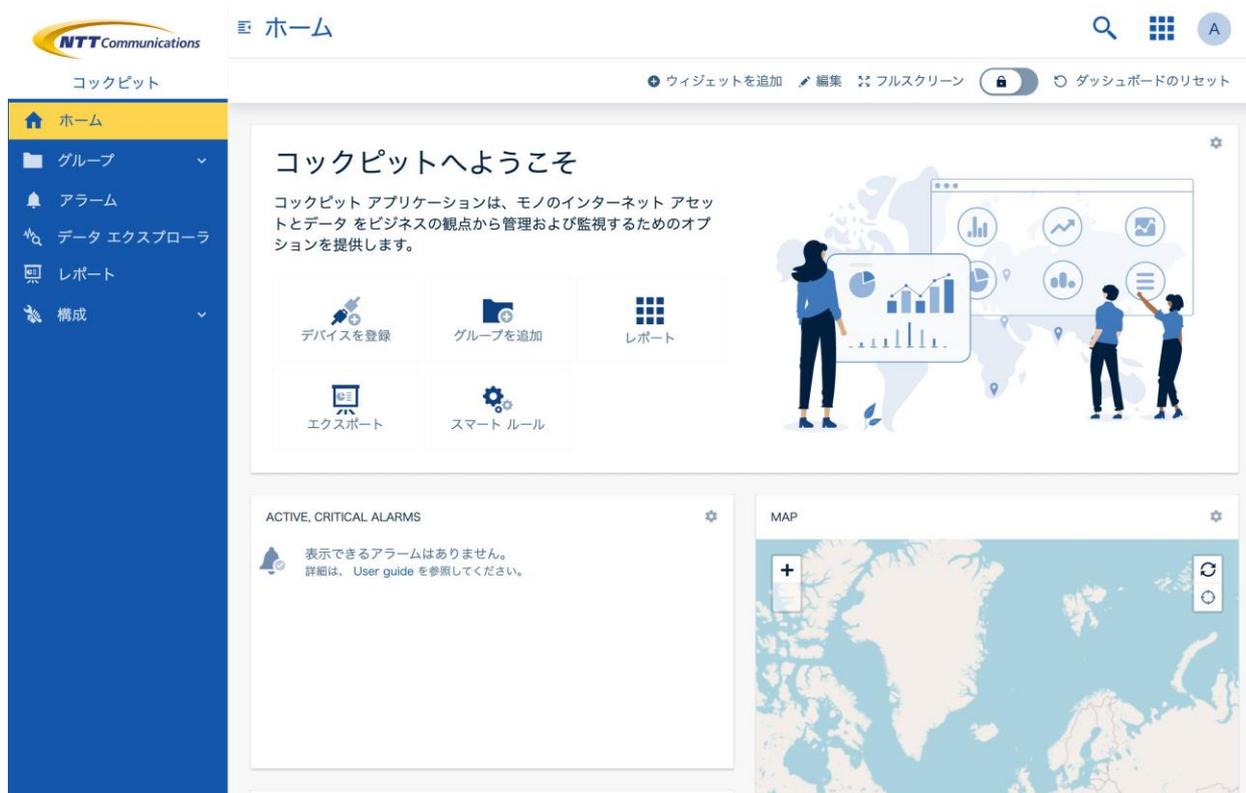
.....

ログイン

[パスワードをお忘れですか?](#)

ログインに成功すると、以下の画面が表示されます。

以下のアプリケーションを「コックピット」と呼び、さまざまなデータの可視化や設定を行う機能があります。



既にThings Cloudにログインしている場合は、画面右上のアプリケーションスイッチャー  をクリックし、「コックピット」アプリを開いてください。

## グループへのデバイス割り当て

### ●グループとは

コックピットアプリでは、作成したデバイスを「グループ」というアセットに割り当てることにより、デバイスのデータをグループ単位で管理することができます。以下の手順で、作成したデバイスをグループに割り当てます。

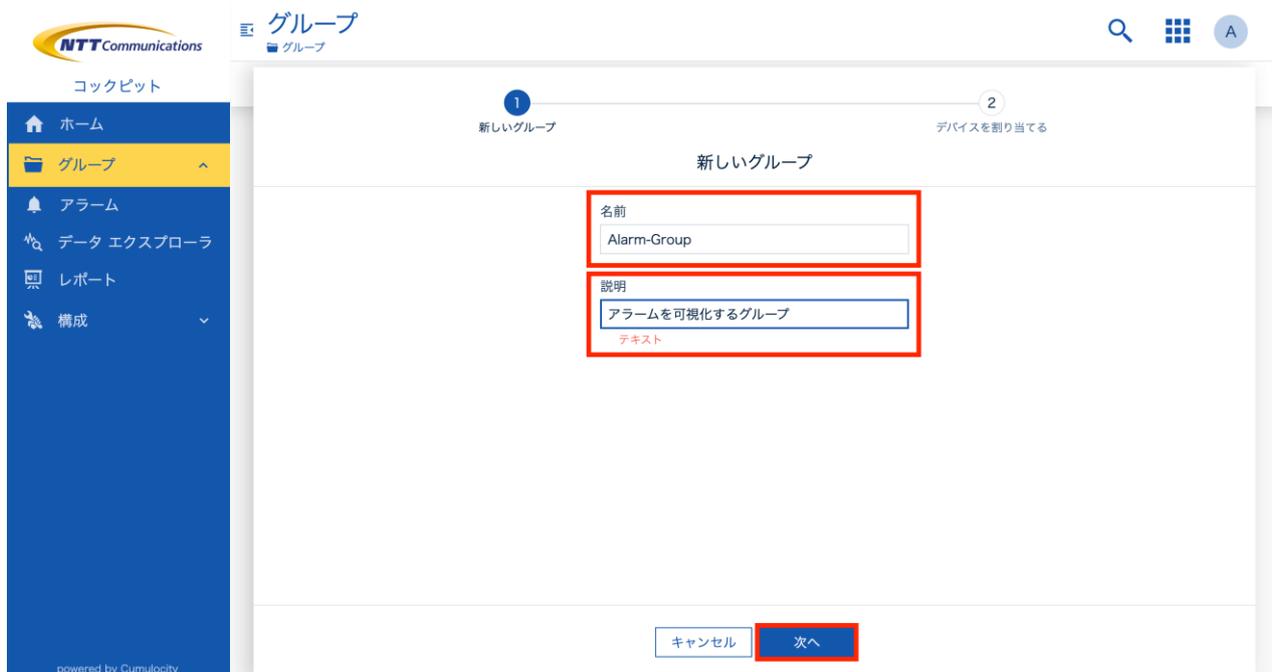
### ●デバイスの割り当て

コックピットアプリの画面左に表示されているナビゲータの中から、「グループ」メニューを選択すると、以下の画面が表示されます。まだグループを作成していないので、何も表示されません。

画面右上の「グループを追加」を選択し、新しいグループを作成します。



グループの名前と、付随する説明を記載し、次へを押下します。



今作成しようとしているグループに対し、扱いたいデバイスを割り当てる画面に遷移します。

画面には、本レシピで作成したデバイスが表示されています。これを割り当てるため、左にあるチェックボックスにチェックを入れ、作成を押下します。

テナントにある全てのデバイスが表示されるため、先ほどNode-REDのMQTT設定時入力したクライアント（デバイスを一意に示すID）と一致するデバイスを選択してください。

クライアントに `Raspberry_Pi001` を設定した場合、デバイス名は `MQTT Device Raspberry_Pi001` になります。

新しいグループ

2 デバイスを割り当てる

デバイスを割り当てる

1 個の選択されたアイテム。 ×キャンセル

ステ...	名前	モデル	シリアル...	グループ	登録日	システム ID	IMEI	アラ
<input type="checkbox"/>	二				2024/04/16 16:48:35	89159		
<input type="checkbox"/>	二				2024/05/09 16:28:46	474388		
<input checked="" type="checkbox"/>	二	MQTT Device Raspberry_Pi001		Alarm-Group	2024/05/16 7:08:01	128220		

キャンセル 戻る 作成

この操作により、作成した自身のデバイスのデータを扱うためのグループが作成されました。

NTT Communications

グループ

サブアセット フィルターなし 1 / 1 アイテム

タイプ ↓ 名前 ↑ モデル アラーム

Alarm-Group

1 - 1 / 1

powered by Cumulocity

## ダッシュボードの追加

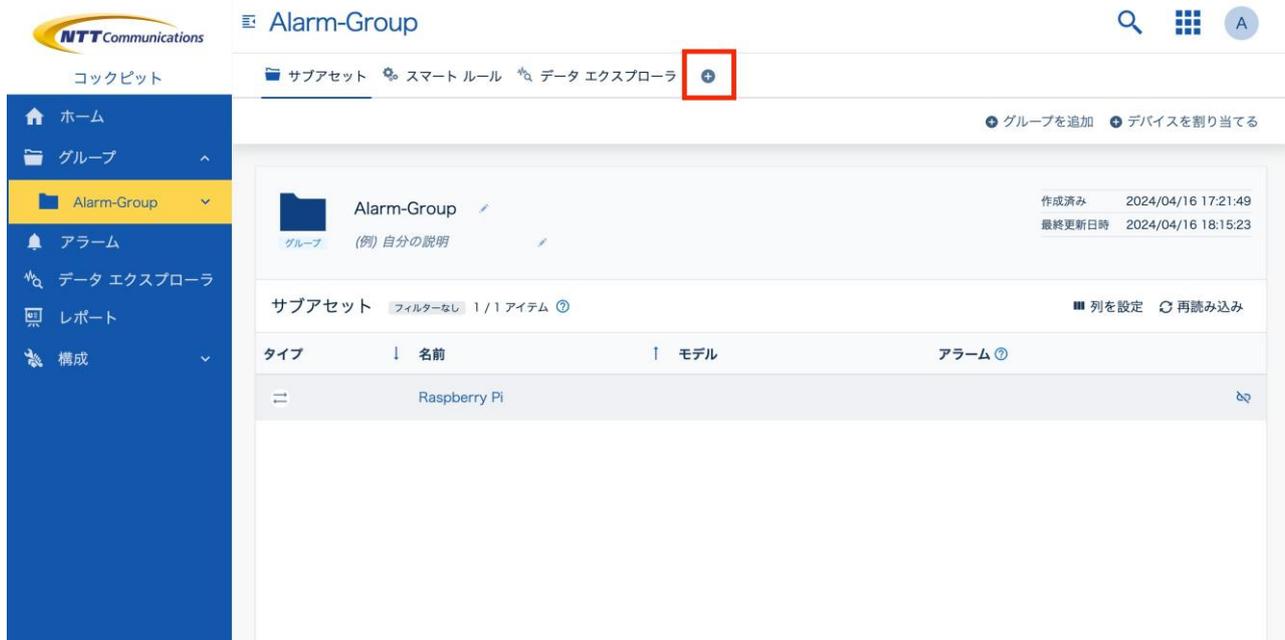
### ●ダッシュボードとは

ダッシュボードとは、グラフやテーブルなどを配置して、データを可視化したり、分析したりするためのツールの1つです。コックピットアプリでは、比較的簡単にダッシュボードを作成し、直感的な画面を作成することが可能です。ダッシュボードの作成方法について、以下で説明します。

### ●ダッシュボードを追加

さっそく、ダッシュボードを追加してみましょう。先ほど作成したグループの名前(例ではAlarm-Group)をクリックし、グループの詳細画面に遷移すると、グループに割り当てたデバイスの一覧が表示されます。

このグループのダッシュボードを作成します。赤枠の+を押下し、ダッシュボード追加画面に遷移します。



The screenshot shows the NTT Communications Cockpit interface. The main content area displays the configuration for an 'Alarm-Group'. The top navigation bar includes a search icon, a grid icon, and a user profile icon. Below the navigation bar, there are tabs for 'サブアセット', 'スマート ルール', and 'データ エクスプローラ'. A red box highlights a '+' button next to the 'データ エクスプローラ' tab. The main content area shows the 'Alarm-Group' details, including the group name, creation time (2024/04/16 17:21:49), and last update time (2024/04/16 18:15:23). Below this, there is a section for 'サブアセット' (Sub-Assets) with a filter set to 'フィルターなし' and 1/1 items. A table lists the sub-assets, with one item 'Raspberry Pi' shown. The table has columns for 'タイプ', '名前', 'モデル', and 'アラーム'.

ダッシュボードを作成するための設定画面が現れます。今回は特に設定を行う必要はありませんので、そのまま保存を押下します。

The screenshot shows the 'Alarm-Group' dashboard configuration interface. A modal window titled 'ダッシュボードを追加' (Add Dashboard) is open, allowing for the configuration of a new dashboard. The interface is divided into several sections:

- 一般 (General):** Includes fields for 'アイコン' (Icon) set to a grid icon, 'メニュー ラベル' (Menu Label) set to 'ダッシュボード', 'ナビゲーション内の位置' (Position in navigation) set to '10000', and 'グローバル ロール' (Global Role) set to 'すべて' (All).
- レイアウト (Layout):** Includes 'テーマ' (Theme) with 'UI を適合' (Fit UI) selected, and 'ウィジェットのヘッダー スタイル' (Widget Header Style) with '標準' (Standard) selected. Other options include '淡色' (Light), '暗色' (Dark), 'ブランディング' (Branding), '分割線' (Separator), '重ね配置' (Overlap), and '非表示' (Hidden).
- プレビュー (Preview):** Shows a preview of the dashboard with a tab labeled 'ダッシュボード' and a placeholder for a widget titled 'ウィジェット タイトル' (Widget Title).
- Buttons:** At the bottom, there are 'キャンセル' (Cancel) and '保存' (Save) buttons. The '保存' button is highlighted with a red border.

画面上部に「ダッシュボード」タブが追加され、ダッシュボードが作成されました。



## アラームウィジェットの追加

### ●ウィジェットについて

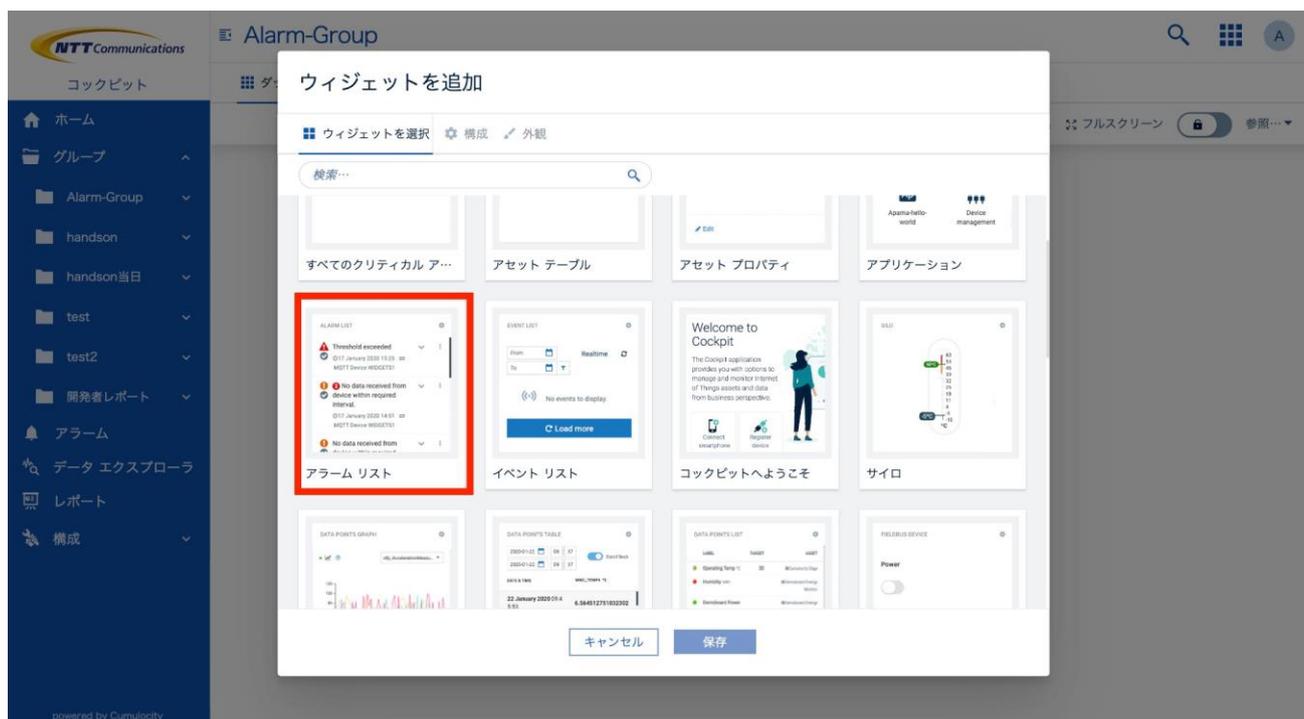
ウィジェットとは、ダッシュボード上に配置することができる、データを可視化するための部品のことです。Things Cloudでは、今回扱うアラームウィジェットの他にも、グラフやテーブルなどさまざまな種類のウィジェットが標準で組み込まれており、自由にダッシュボードをカスタマイズすることが可能です。この章では、今回設定するアラームウィジェットの追加方法について説明します。

## ●アラームウィジェットを追加

グループに紐づくダッシュボードを作成したので、ダッシュボードにアラームウィジェットを追加します。アラームウィジェットは、デバイスから送信されたアラームを可視化するためのウィジェットです。

さっそく、先ほど作成したダッシュボードの画面の中央にある「ウィジェットを追加」を押下すると、ダッシュボードに追加可能なウィジェットの一覧が表示されます。

「アラーム リスト」を探してください。これがアラームウィジェットを指します。このウィジェットを選択してください。



どのようなデータを表示させるかといった設定の画面が表示されます。

以下の表に、設定項目の一覧を示します。

設定項目	内容	詳細
アセットの選択	アラームを表示させたいデバイス	今回は、作成したグループ全体を指定します。これにより、グループに複数のデバイスが割り当てられていた場合でも、全てのデバイスのアラームを一覧で表示することが可能です。
タイトル	ウィジェットのタイトル	任意のタイトルを入力します。
ステータス	表示するアラームのステータス	今回は、「クリア済み」のみチェックボックスから外します。これにより、対処が完了したアラームをウィジェットから非表示にすることが可能です。
タイプ	表示するアラームの種類	入力欄を空欄に設定することで、全てのアラームを表示させます。

設定できたら、保存を押下します。



これで、ダッシュボードにアラームウィジェットが追加されました。



## アラームウィジェットでアラームを可視化

今作成したアラームウィジェットでアラームが可視化できるかどうかを確認するために、まずはデバイスからアラームを送信してみましょう。

先ほど作成したNode-REDのフロー画面に戻り、アラームを送信するノード「緊急停止アラーム送信」があることを確認します。ノードの左側にあるボタンをクリックし、アラームを送信します。

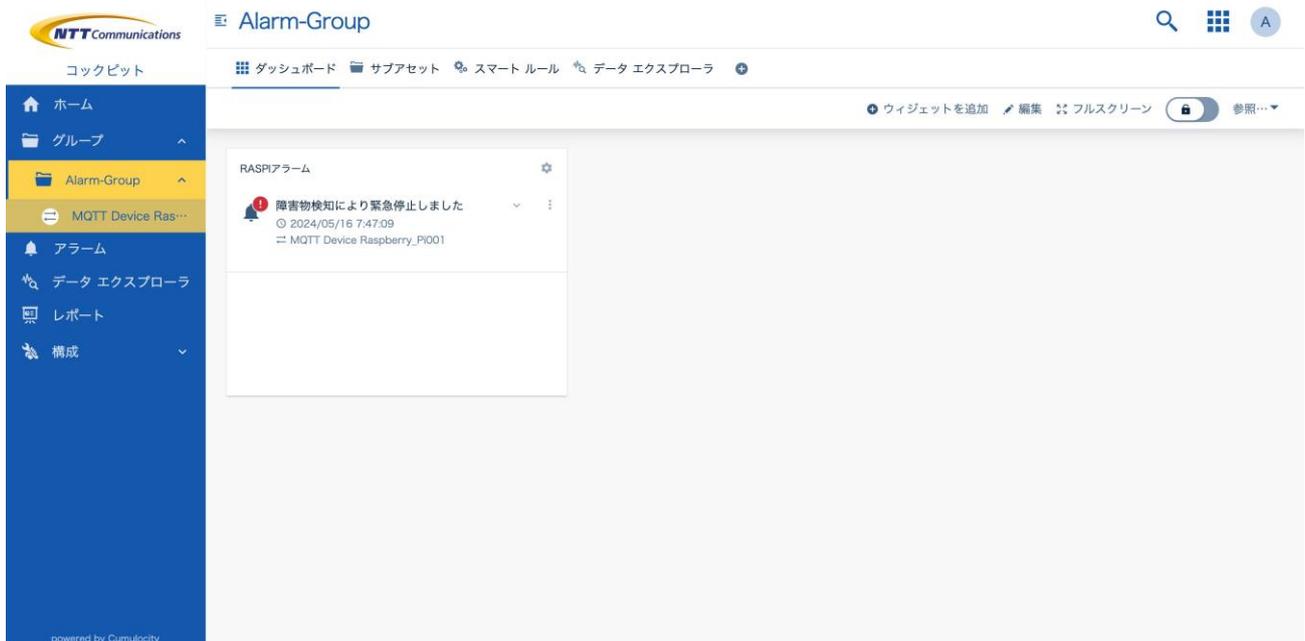
先ほどデバイス登録のために送信したアラームが残っている場合は、そのまま再度送信しても問題ありません。ただし、同じアラームを連続で送信してもアラームが複数表示はされません。

※クリアされる前に同じアラームタイプのアラームを送信すると、アラームのカウント数が増加します。

アラームを送信すると、先ほど作成したウィジェットの中にデバイスから受信したアラームが表示されます。

ウィジェットの中には、デバイスから送信されたアラームの一覧が表示されています。

アラームを選択すると、アラームの詳細情報を表示することができます。さらに、承認ボタン  を押すと、対処したアラームに対し確認済みのステータスに変更することができ、クリアボタン  を押すと、アラームをクリアすることができ、ウィジェットから非表示になります。



## スマートルールを使用したメール通知の設定

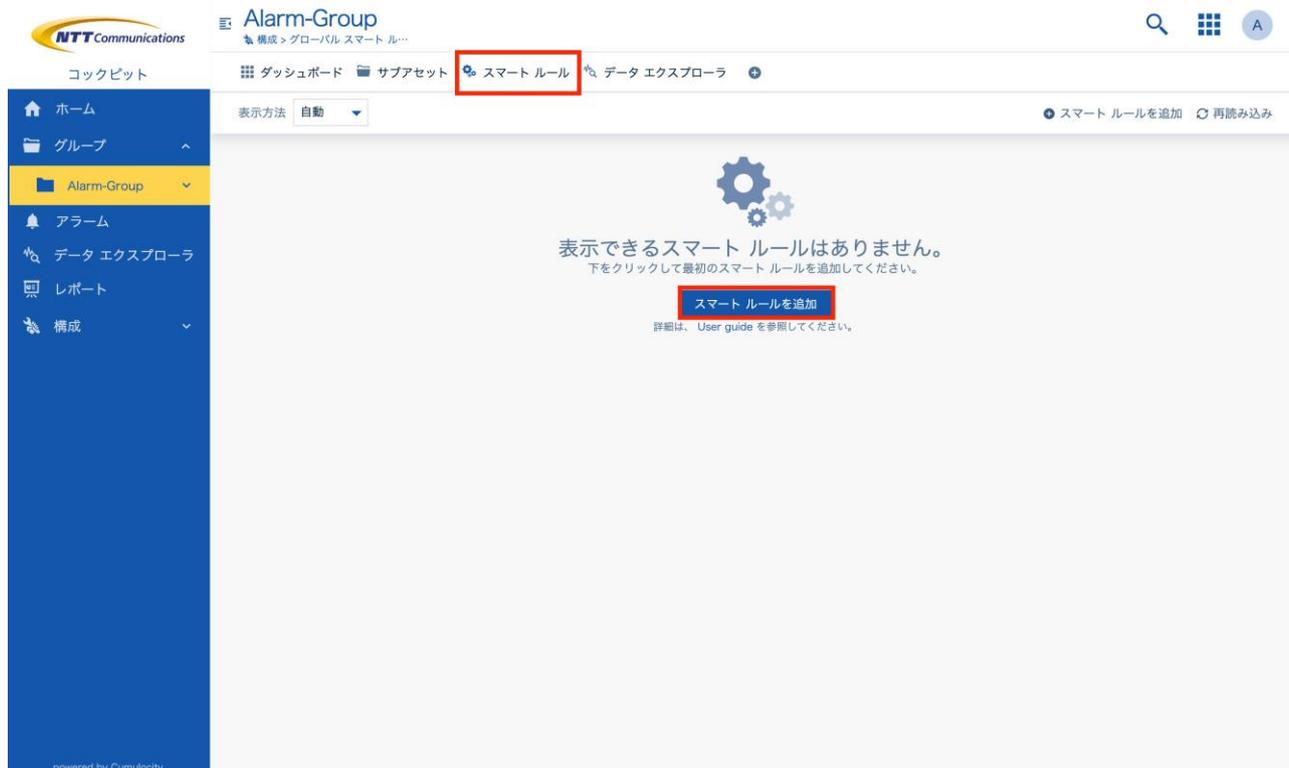
### スマートルールとは

Things Cloudでは、スマートルールという機能が提供されています。スマートルールとは、デバイスから送信されたデータをリアルタイムに処理し、自動的にさまざまなアクションを行うための機能です。例えば、デバイスからのデータが指定の値を超えた場合に、メール通知を行うような処理を自動化することができます。

コックピットアプリには、簡単にスマートルールを作成するためのテンプレートが多数用意されており、今回もテンプレートを使用してスマートルールを設定していきます。

### メール通知の設定

デバイスからアラームを受信したら、メールで通知を行うというスマートルールを設定してみましょう。先ほどアラームウィジェットを作成したダッシュボードの画面の上部に表示されている「スマートルール」タブを選択すると、スマートルールを追加する画面に遷移します。画面中央の「スマートルールを追加」を押下し、スマートルールの設定画面に遷移します。



スマートルールのテンプレートが表示されます。たくさんのテンプレートを用意しているため、柔軟にルールを設定しさまざまなアクションを自動化できることがわかります。今回は「アラーム時に電子メールを送信」というテンプレートを選択します。



テンプレートを選択すると、スマートルールでどのようなアラームに対しどのような電子メールをどこに送信するかといった設定の画面が表示されます。

以下の表に、設定項目の一覧を示します。

設定項目	内容	詳細
ルール名	スマートルールの名前	任意の名前を入力します。
次のタイプのアラームの場合	どのようなアラームに対してメール通知を行うか	今回は、障害物の検知による緊急停止をしたというアラームに対してメール通知を行うように設定します。 入力欄には、緊急停止アラームのタイプである "com_emergencyStopAlarm" と入力します。 まず「アラームタイプを追加」を押下することで、複数のアラームの種類を指定することも可能です。
Eメールを送信	システムから送信されたメールの返信先の設定	送信先をはじめ、CCやBCC、システムから送信されたメールの返信先の設定を行うことができさらに、メールの件名や本文の設定も行うことが可能です。 件名や本文にデフォルトで指定されているプレースホルダーについては、以下の内容が反映されます。 #{source.name} : アラームを発生させたデバイスの名前 #{severity} : アラームの重要度

		# {text} : アラームの内容
対象のアセットもしくはデバイス	どのデバイスのアラームに対してメール通知を行うか	今回は、作成したグループ全体を指定します。

アラームタイプとは、デバイスから送信されるアラームの種類のことを指します。1種類のアラームに対し、1つのアラームタイプが紐づきます。

一連の設定が完了したら、保存を押下します。

新しいグローバル スマート ルール



アラーム時に電子メールを送信

\* マークのついたフィールドは必須です。

✔ 有効

**1 ルール名**

警備ロボットの緊急停止通知

**2 次のタイプのアラームの場合:**

com\_emergencyStopAlarm -

+ アラーム タイプを追加

### 3 エメールを送信:

送信先

(例) taro.yamada@example.com,tatsuki.

- i** 1つのアドレスまたはカンマで区切って複数のアドレスを入力してください。

送信先(CC)

(例) taro.yamada@example.com,tatsuki.

- i** 1つのアドレスまたはカンマで区切って複数のアドレスを入力してください。

送信先  
(BCC)

(例) taro.yamada@example.com,tatsuki.

- i** 1つのアドレスまたはカンマで区切って複数のアドレスを入力してください。

返信先

(例) taro.yamada@example.co.jp

#{source.name} からの新しい#{severity} アラーム

#{source.name} から新しい #{severity} アラームを受信しました。アラーム テキストは "#{text}" です。

### 4 対象のアセットもしくはデバイス

デバイス名またはデバイスのプロパティ値



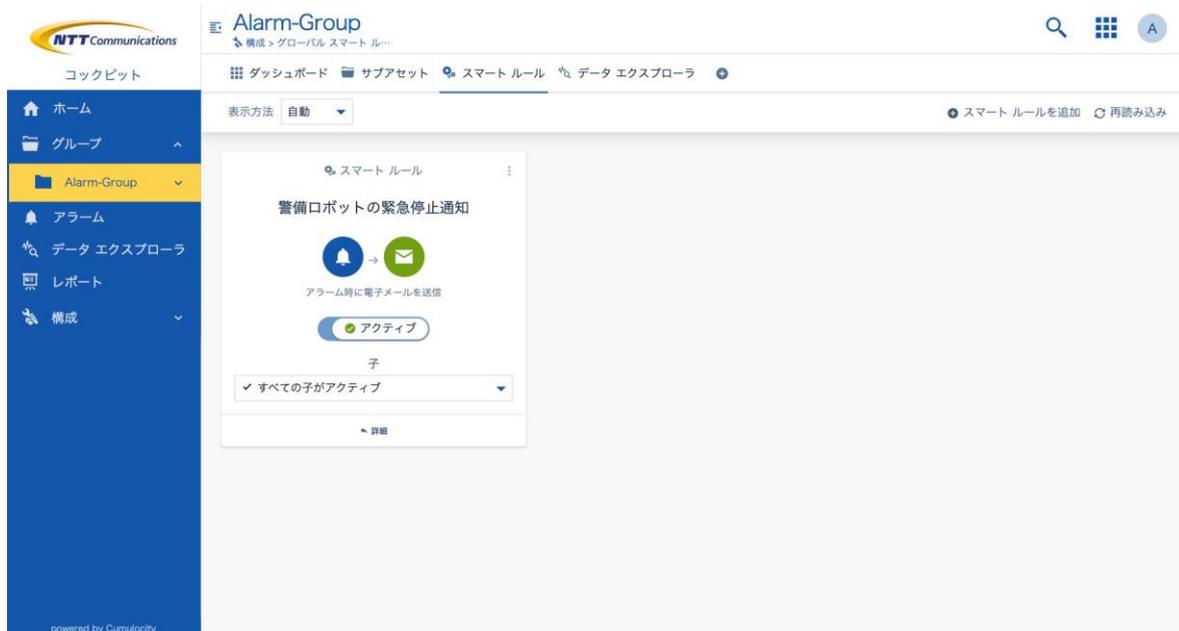
Alarm-Group

1 子アセットにも有効化

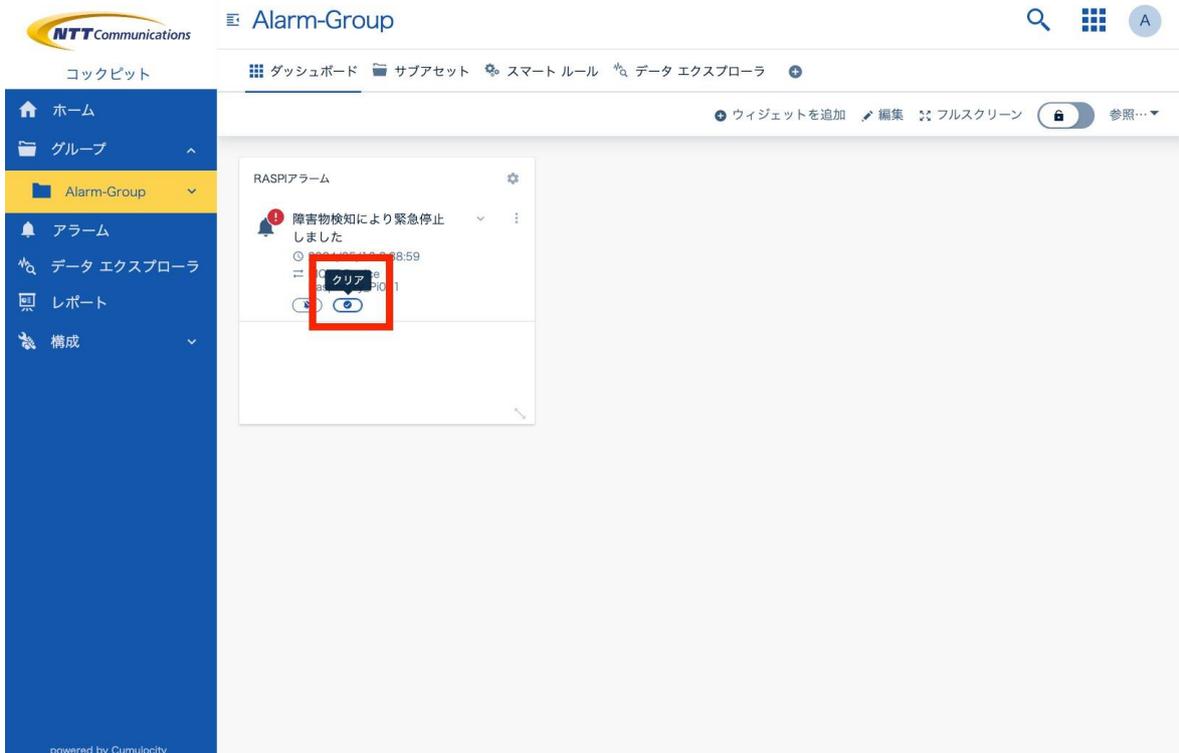
キャンセル

作成

これで、デバイスからアラームを受信したら、メールで通知を行うというスマートルールを設定することができました。



では、改めてもう一度アラームを発生させ、メールで通知されるかを確認してみましょう。この時、先ほど受信したアラームをクリアするようにしてください。赤枠の  を押下することで、アラームをクリアすることができます。スマートルールは、複数回同じ種類のアラームを受信した場合、最初の一度のみ作動します。そのため、先ほど試しに発生させたアラームが残っていると、正しくメールが送信されません。



では、Node-REDの画面からもう一度緊急停止アラームを発生させてみましょう。すると、まもなくして指定したメールアドレスに対しメールが届きます。



これで、デバイスから発生したアラームをダッシュボードで可視化し、さらにアラームが発生した場合にはメールで通知を行う設定ができました。

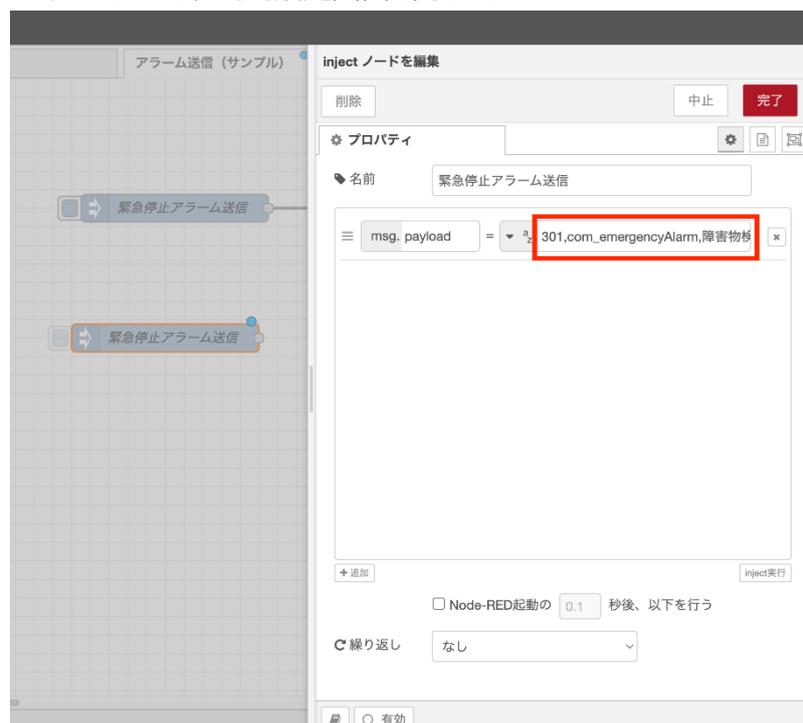
## 【応用編】独自のアラーム内容を送信

最初のアラームが送信できThings Cloud上で確認ができたなら、次はアラームの内容を変更してみましょう。送信する際のデータ内容を変更することで、アラームの種類や重大度、アラームのテキストを自由に設定することが可能です。

### アラームの内容を変更

別のアラームを送信するための手順：

1. Node-REDのフロー画面に戻り、アラームを送信するノード「バッテリーアラーム送信」を右クリックし、「ノードをコピー」を選択します。
2. 次にフロー上で右クリックをして「ノードを貼り付け」を選択し、新たにアラームを送信するノードを配置します。配置できたら、そのノードをダブルクリックして設定画面を開きます。



3. 設定画面で、アラームの内容を変更します。msg.payload の右に設定されているのが送信されるデータの内容です。以下の表にデータの仕様を示しているので、自由に変更してみましょう。

※ Things CloudのMQTT通信では、通常SmartRESTというプロトコルを用いてデータを送信します。  
SmartRESTでは、あらかじめ定義されたカンマ区切りのフォーマットに合わせてデータを送信します。

データ例：

304,deviceBatteryAlarm,機器のバッテリー残量が低下しています

#	データ項目	内容	詳細
1	SmartRESTのテンプレートID	301 (クリティカルアラーム) 302 (メジャーアラーム) 303(マイナーアラーム) 304 (警告アラーム)	SmartRESTのテンプレートを示す番号 標準のテンプレートがあらかじめ定義されており、アラームの場合は重大度に応じて4種類が定義 ※ 独自のテンプレートを作成することも可能
2	アラームタイプ	deviceBatteryAlarm deviceFailureAlarm など	アラームの種類を示す文字列任意の文字列を設定可能
3	アラームテキスト	アラームの詳細情報	アラームの内容を示す文字列（可視化画面に表示される）任意の文字列を設定可能

変更例：

301.deviceFailureAlarm,機器が故障しました

### inject ノードを編集

削除 中止 完了

プロパティ

名前 バッテリーアラーム送信

msg. payload = a\_z 304,deviceBatteryAlarm,機器のバッ

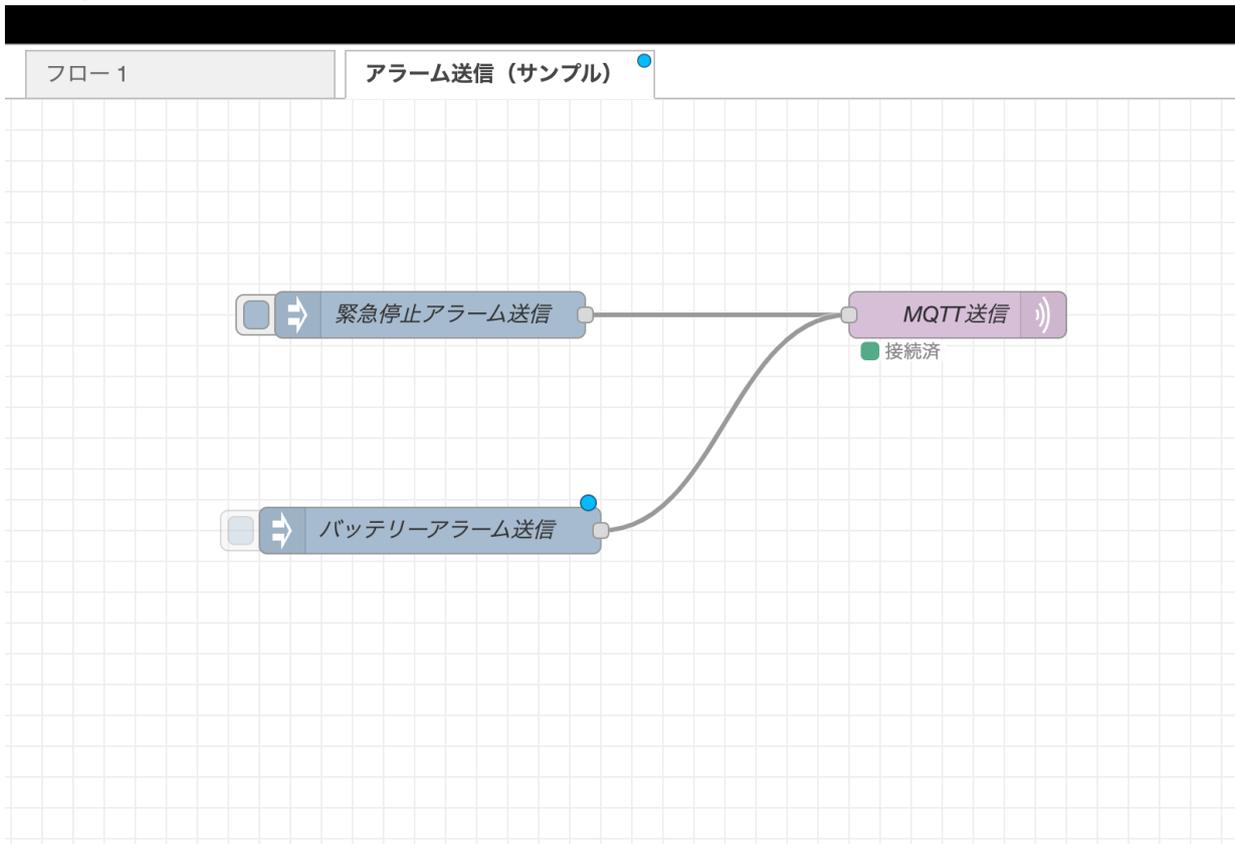
+追加 inject実行

Node-RED起動の 0.1 秒後、以下を行う

繰り返し なし

有効

- 設定が完了したら、画面右上の「完了」をクリックして設定を保存します。次に、編集した新しいノードを「MQTT送信」ノードに接続してください。



- 新しいノードが追加できMQTT送信ノードに接続されたら、画面右上の「デプロイ」をクリックして設定を反映させて完了です。

### SmartRESTとは

SmartRESTは、IoTデバイスからThings Cloudにデータを送信するためのプロトコルの一つです。REST APIのようにThings Cloudに対して各種データを送信することができますが、データの形式や送信方法がREST APIとは異なります。SmartRESTでは、あらかじめ定義されたテンプレート（ユーザーが定義することも可能）に基づいてカンマ区切り（CSV）のデータを送信することで、アラームや計測値などさまざまなIoTデータを簡単に送信することができます。詳しくは[こちら](#)をご参照ください。

### 追加したアラームを送信

新しいアラームの送信設定ができたので、再度デバイスからアラームを送信してみましょう。先ほど作成したノードの左のボタンをクリックしてアラームを送信します。

アラームを送信したら、再びThings Cloudのダッシュボードに戻ります。先ほど送信したアラームとは異なるアラームがウィジェットに表示されることが確認できるはずです。

このようにデバイスから任意のアラーム内容を送信ことができ、Things Cloud上でアラームの可視化を行うことが可能です。



コックピット

## Alarm-Group

ダッシュボード サブアセット スマート ルール データ エクスプローラ

- ホーム
- グループ ^
- Alarm-Group ^
- MQTT Device Ras...
- アラーム
- データ エクスプローラ
- レポート
- 構成 v

### RASPIアラーム

機器のバッテリー残量が低下して  
います

2024/05/16 7:54:51

MQTT Device Raspberry\_Pi001

## 3.最後に

### 3-1.次のステップ

本レシピでは、Raspberry PiをIoTデバイスとして利用し、Things Cloudを用いてデバイスから受信したアラームを可視化、さらにスマートルールを用いてメールで通知する方法について解説しました。

Things Cloudを利用することで、デバイスの監視や運用を効率化することができるということが少しだけ実感できたのではないのでしょうか。

今回はIoTデバイスから送信される「アラーム」について扱いましたが、デバイスからは他にも「メジャーメント(計測値)」「イベント(デバイスの正常な状態を知らせる要素)」などが送信されています。次のステップとして、アラーム以外のデータをThings Cloud上で扱う方法についてチャレンジしてみましょう。

### 3-2.Things Cloudの詳細情報とお申し込みについて

ここまで読んでいただき誠にありがとうございました。

Things Cloudのさらなる詳細情報については、[Things Cloud紹介ページ](#)をご覧ください。

資料請求に関しては、[資料請求ページ](#)からお願いいたします。