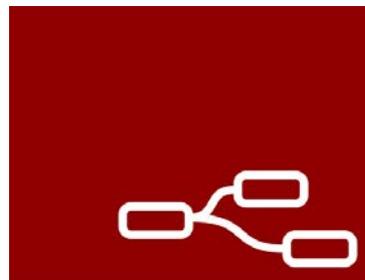


抜粋版

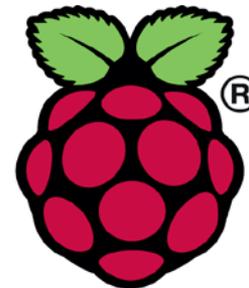
AWS IoTを使用したサーモセンサ開発キット ～食品輸送、人検知、機械故障等の温度管理まで幅広く応用～ 実践編



AWS IoT Core



Node-RED



Raspberry Pi

スペクトラム・テクノロジー株式会社

<https://spectrum-tech.co.jp>

sales@spectrum-tech.co.jp

サーモセンサ開発キット 目次

Pi運用マニュアル

1. RaspberryPiについて
2. Linux基本コマンド
3. 基本操作
4. 日常運用(ウイルススキャン、更新)

AWS IoT開発

- ① メニュー
- ② AWS設定
- ③ AWS IoT設定
- ④ サーモセンサ情報収集、アラーム送付
 - 全体構成
 - サーモセンサ概要
 - サーモセンサーRaspberryPi接続
 - サーモセンサ単体試験
 - サーモセンサ画像試験
 - Mqttプログラム設定
 - AWS IoT折り返し試験
 - リアルタイム表示(グラフ)
 - IoTイベント設定
 - IoTイベント試験
 - 遠隔リアルタイムモニタ
 - CSV取り出し
 - CSV画像再生

ページ

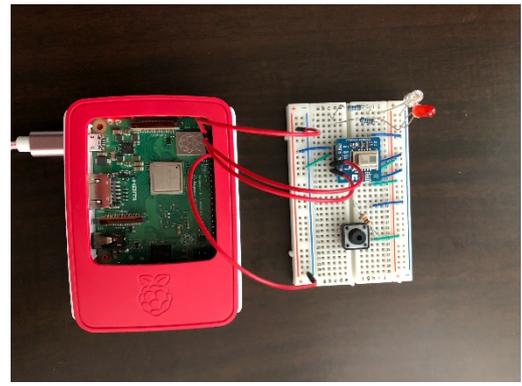
- [3](#)
- [4](#)
- [5](#)
- [6](#)

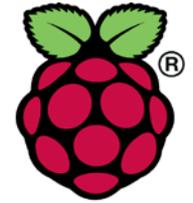
抜粋版のため本文と一致しません

ページ

- [8](#)
- [9](#)
- [16](#)

- [29](#)
- [30](#)
- [31](#)
- [32](#)
- [33](#)
- [34](#)
- [36](#)
- [37](#)
- [39](#)
- [47](#)
- [50](#)
- [51](#)
- [53](#)





Pi運用マニュアル

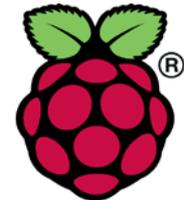
1. Raspberry Piについて

既に全世界で1000万台以上販売された手のひらサイズのコンピュータです。
LinuxベースのRasbianOSで動作しております。

2. Linux基本コマンド

① システム関係

- 起動: 電源を入れると自動で起動します。
- 再起動: # reboot
又は、アプリケーション>ログアウト>再起動; 左上のメニューから
- 終了: # shutdown
又は、アプリケーション>ログアウト>シャットダウン; 左上のメニューから
- ログアウト # logout
又は、アプリケーション>ログアウト>ログアウト; 左上のメニューから
- **日本語／英語の入力切替**: キーボードのctl+jを同時に押します。又は右上のアイコン(右から7個目)からプルダウンで選択



Pi運用マニュアル

2. Linux基本コマンド

② ディレクトリ操作、コピー、移動、削除

root@:~# **cd** /root/Documents ディレクトリの切り替え
 root@:/root/Documents# **ls** ファイルとディレクトリの表示(表示したら操作したいファイルを右クリックでコピーして操作します)

root@:~# **cp** ファイル名 ディレクトリ 配下のディレクトリのファイルを別のディレクトリへコピー
 root@:~# **mv** ファイル名 ディレクトリ 配下のディレクトリのファイルを別のディレクトリへ移動
 root@:~# **rm** ファイル名 ファイルの削除

便利な機能 **rm -help** コマンドのオプションが分からない場合は、ヘルプで問い合わせる。すべてのコマンド共通(マイナスを2個とhelp)

③ ユーザ権限、プロセス他

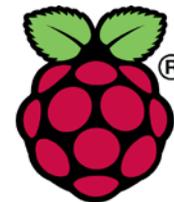
root@:~ \$ **su -** スーパーユーザ(root)に切り替え、パスワードを入力
 root@:~# **ps a** 現状の動いているプロセスを表示
 root@:~# **kill** 特定のプロセスを強制終了

root@:~# **apt-get install pkg** パッケージのインストールなどに使用
 root@:~# **date** 日付、時間の設定を行います。

root@:~# **leafpad** /etc/network/interfaces インタフェースに記述している内容を変更します。Viよりも使いやすいです。

④ モジュール、usb、メモリ、HDDなどの表示

root@:~# **lsmod** linuxのモジュールリスト表示
 root@:~# **lsusb** usbのデバイス表示
 root@:~# **free -mt** メモリ使用状態表示
 root@:~# **df -h** HDD(マイクロSD)の使用状態表示



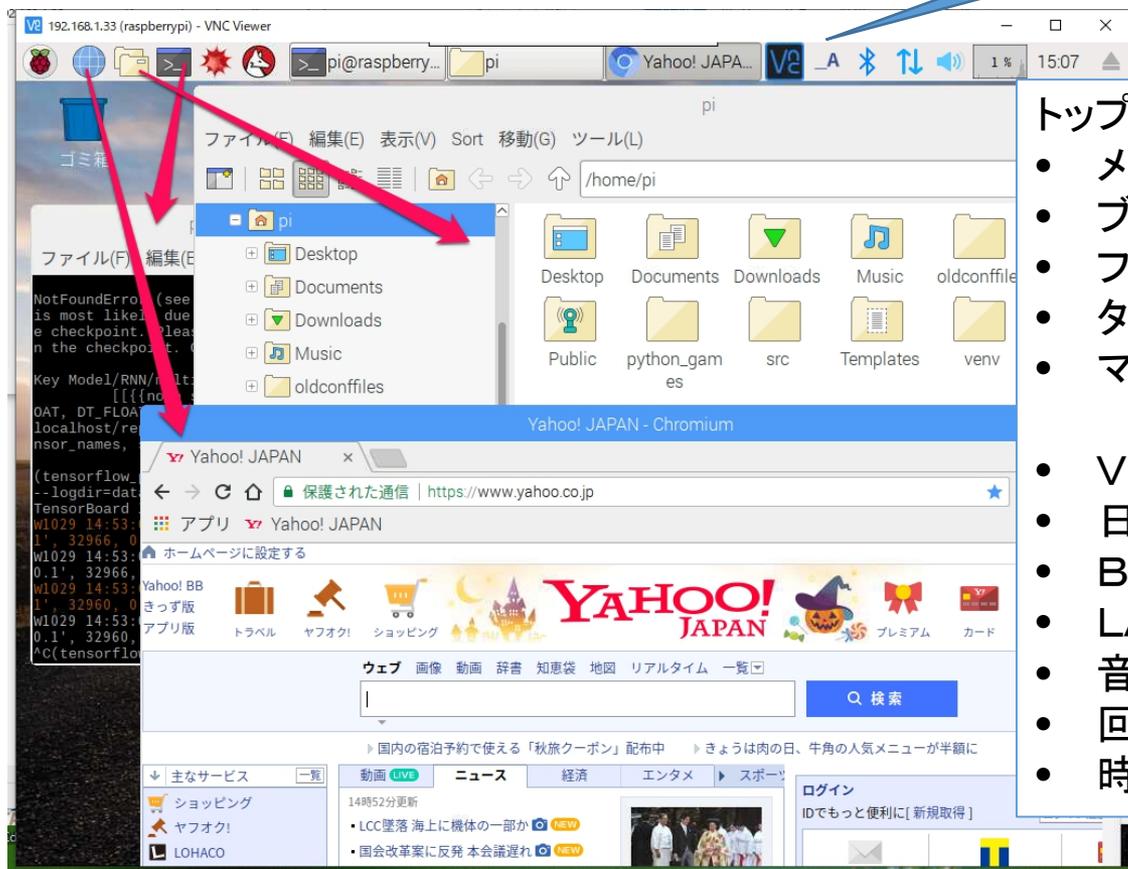
Raspberry Pi

Pi運用マニュアル

3. 基本操作

① 表示画面と内容

Anthyが出ない場合は、一度 Japaneseを選択後、再度 Anthyを選択してください。



トップ画面(上段のタスクバーで選択)

- メニュー
 - ブラウザ
 - ファイルマネージャ
 - ターミナル
 - マルチ画面選択
-
- VNC
 - 日本語入力
 - BLE
 - LAN/WiFi
 - 音量
 - 回線効率
 - 時刻

AWS IoT開発

②. AWS設定

- 手順1: アカウントの作成
 - <https://aws.amazon.com/jp/register-flow/>
 - 必要なメールアドレス、パスワードなどを入力します。



AWS アカウント作成の流れ

AWS アカウントを作成すると、1年間の無料利用枠はもちろん、AWS クラウドの世界中のリージョンで提供されるすべてのサービスを始めることができます。こちらでは日本のお客様にAWS アカウント作成におけるポイントをご紹介します。

今すぐクラウドを無料でお試ください

[今すぐ無料アカウント作成 >](#)

12 か月間の AWS 無料利用枠と、24 時間年中無休のカスタマーサービスやサポートフォーラムなどの AWS の基本的なサポート機能を利用できます。

ステップ 1: AWS アカウントの作成



このページの上部タイトルおよび、末尾に設置されているオレンジ色のアカウント作成ボタンよりサインアップ画面へ移動します。

各ページ右上 ① の言語選択ボックスより、「日本語」でない場合「日本語」を選択後、こちらのサインアップ画面へお進みください。

- 最初に AWS アカウントとなる情報を設定します。
- ② の「E メールアドレス」には、AWS へのログインに利用したいメールアドレスを設定します。(※)
 - ③ の「パスワード」および「パスワードの確認」で AWS へのログイン時に使用するパスワードを設定し、さらに確認用にもう一度同じパスワードを入力します。
 - ④ の「AWS アカウント名」テキストボックスに、お客様のお名前を半角アルファベットで入力します。
 - 入力後、「続行」ボタンをクリックします。

※クリックすると大きな画像でご覧いただけます。

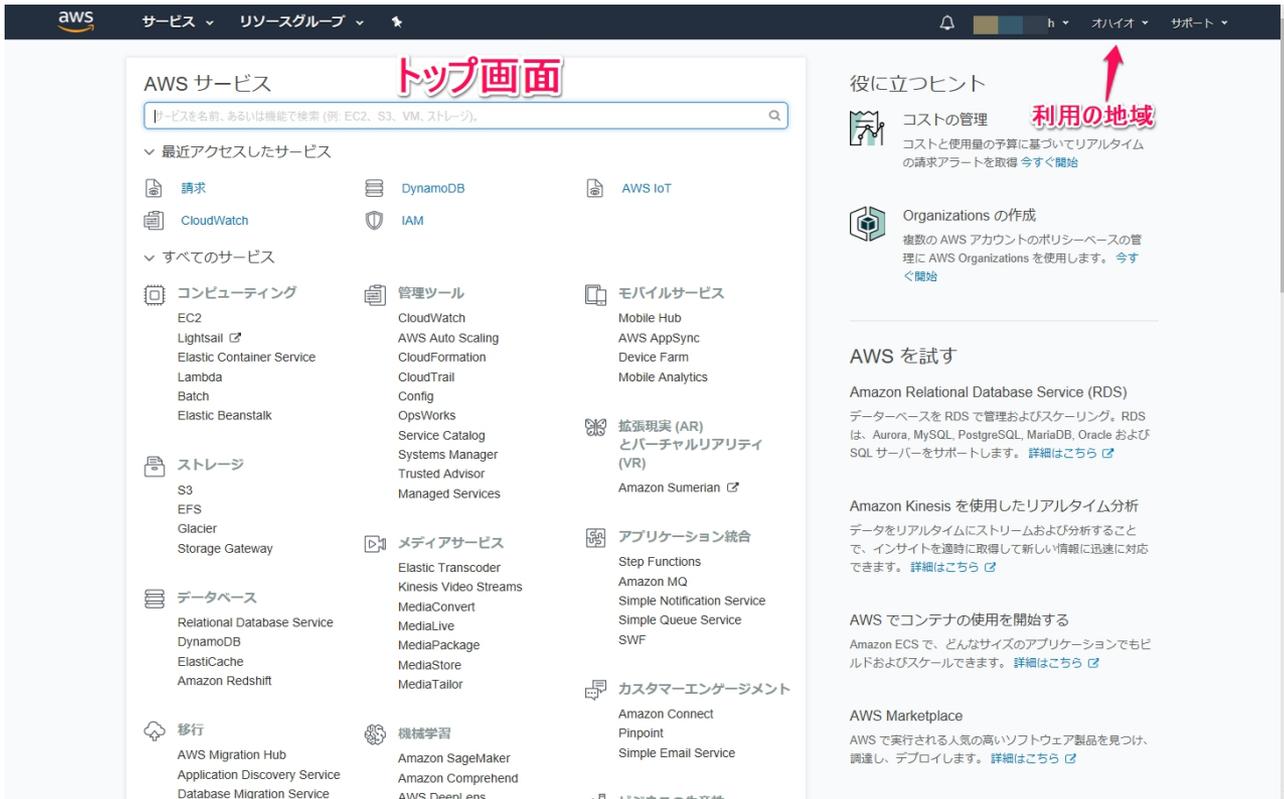
※登録いただくメールアドレスは、AWS 側からの通知等にも利用されます。複数の方へ

AWS IoT開発

②. AWS設定

- AWSトップ画面

- 利用する場合に、地域を意識して設定してください。地域毎に料金が変わったり、利用できるサービスが限定されている場合があります。



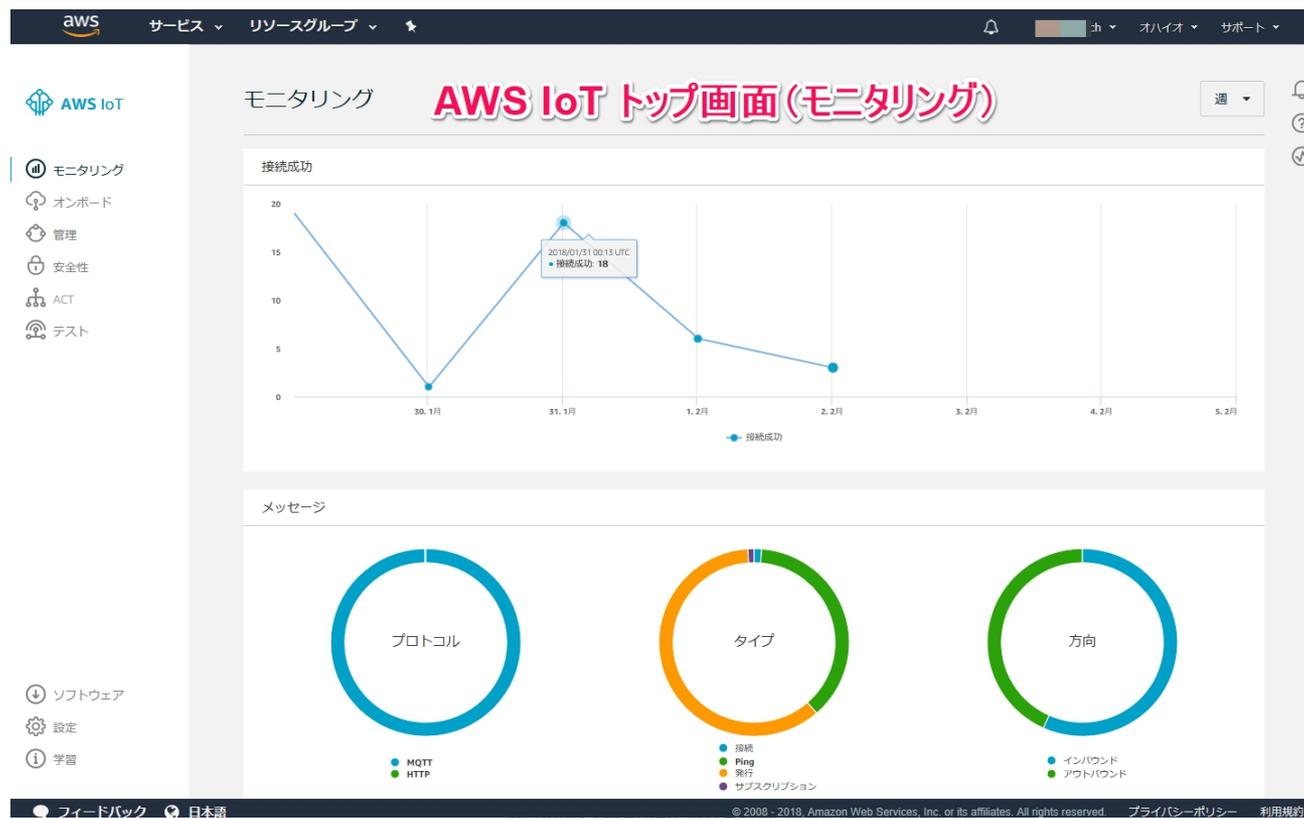


AWS IoT開発

③. AWS IoT設定

● 画面説明1

- AWSTップ画面からAWS IoTのサービスを選択します。
- モニタリング: 接続状況がひとめで分かります





AWS IoT開発

③. AWS IoT設定

- 画面説明2
 - オンボード: 特に使用しません
 - 管理: モノ

オンボード

AWS IoTに接続する

デバイスの設定

AWS IoT Device SDKの接続ウィザードを使用してデバイスまたはコンピュータをAWS IoTに接続します。

AWS IoT ボタン

AWS IoT ボタンは、ボタン1押しでAWS IoTにメッセージを送信する単一用途のデバイスです。

モノ

AWS IoTに接続するモノを登録します。

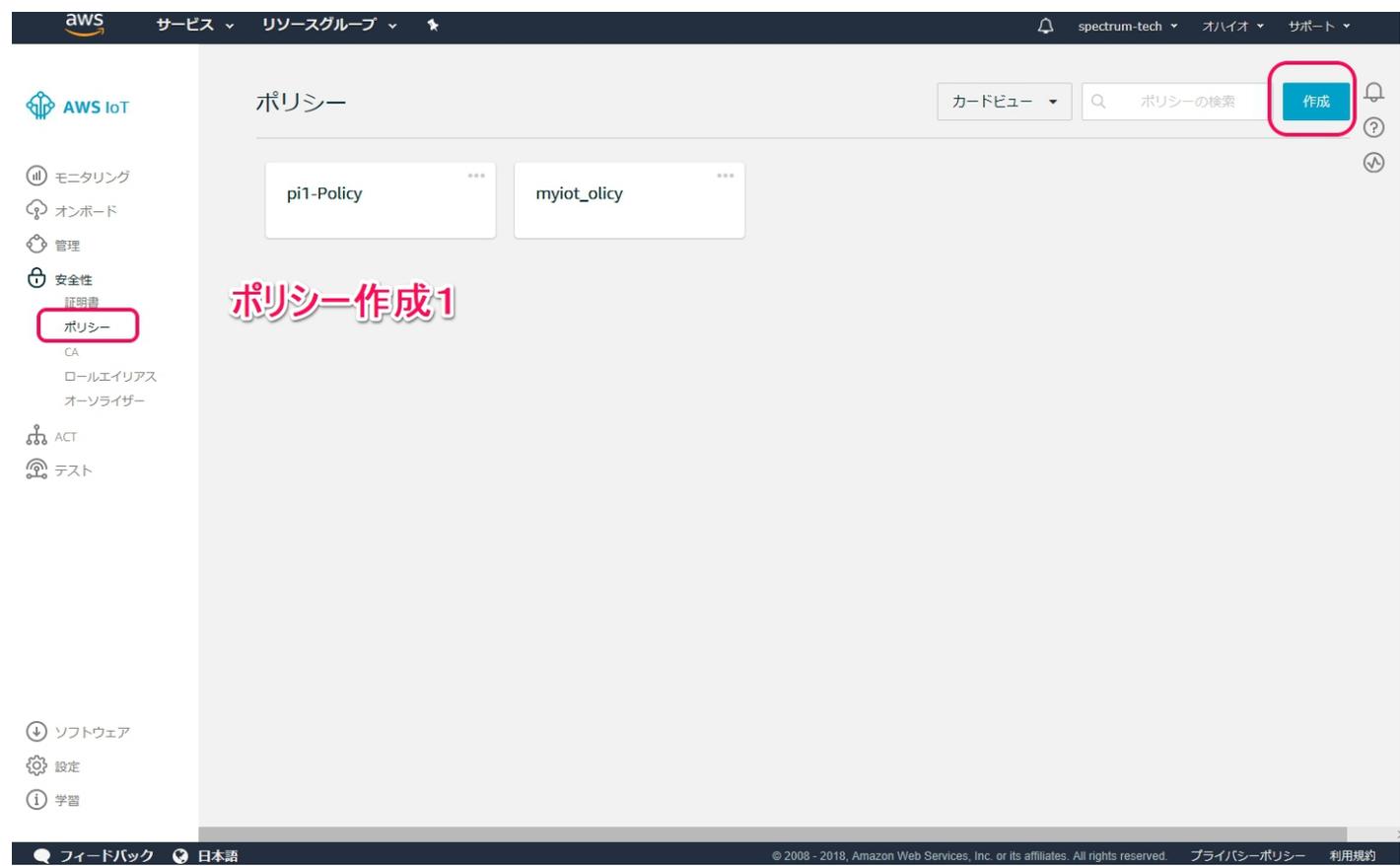
モノに証明書、ポリシーを割り付けます。



AWS IoT開発

③. AWS IoT設定

- ポリシー作成1
 - 安全性 > ポリシーの作成を押します。

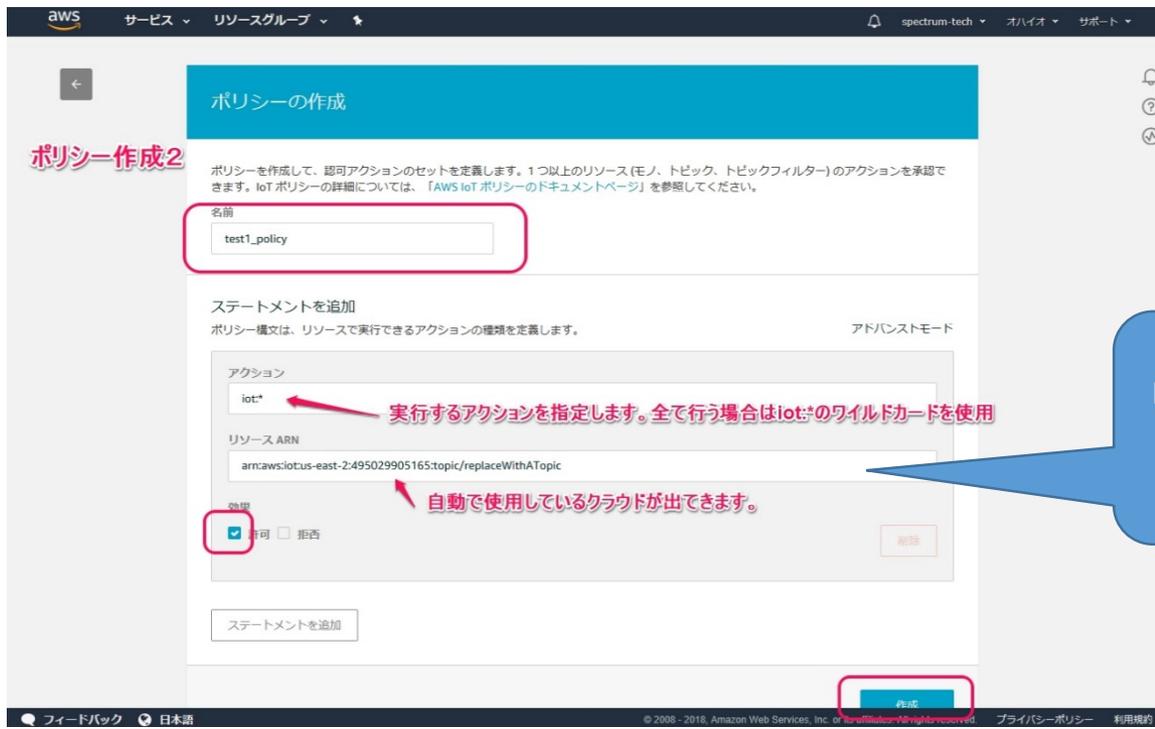


AWS IoT開発

③. AWS IoT設定

• ポリシー作成2

- 名前を入力します。
- アクション: `iot:connection`などの実行するアクションを入れます。ワイルドカードも使用できます。慣れるまで、`iot:*`と入力します。
- リソースARN: 現在利用しているクラウドのリソースが表示されます。そこからの接続のみに限定されます。又は*の無制限でも可能です。



aws サービス リソースグループ spectrum-tech オハイオ サポート

ポリシーの作成

ポリシー作成2

ポリシーを作成して、認可アクションのセットを定義します。1つ以上のリソース(モノ、トピック、トピックフィルター)のアクションを承認できます。IoTポリシーの詳細については、「AWS IoT ポリシーのドキュメントページ」を参照してください。

名前
test1_policy

ステートメントを追加
ポリシー構文は、リソースで実行できるアクションの種類を定義します。 アドバンスドモード

アクション
iot* **実行するアクションを指定します。全て行う場合はiot:*のワイルドカードを使用**

リソース ARN
arn:aws:iot:us-east-2:495029905165:topic/replaceWithATopic **自動で使用しているクラウドが出てきます。**

確認
 許可 拒否

ステートメントを追加

作成

フィードバック 日本語 © 2008 - 2018 Amazon Web Services, Inc. or its affiliates. All rights reserved. プライバシーポリシー 利用規約

リソースARNを*
にしてください。

AWS IoT開発

③. AWS IoT設定

• ポリシー作成3

- 安全性 > 証明書 > モノの登録で作成した証明書にポリシーをアタッチします。
- 右クリックして > ポリシーのアタッチを押します



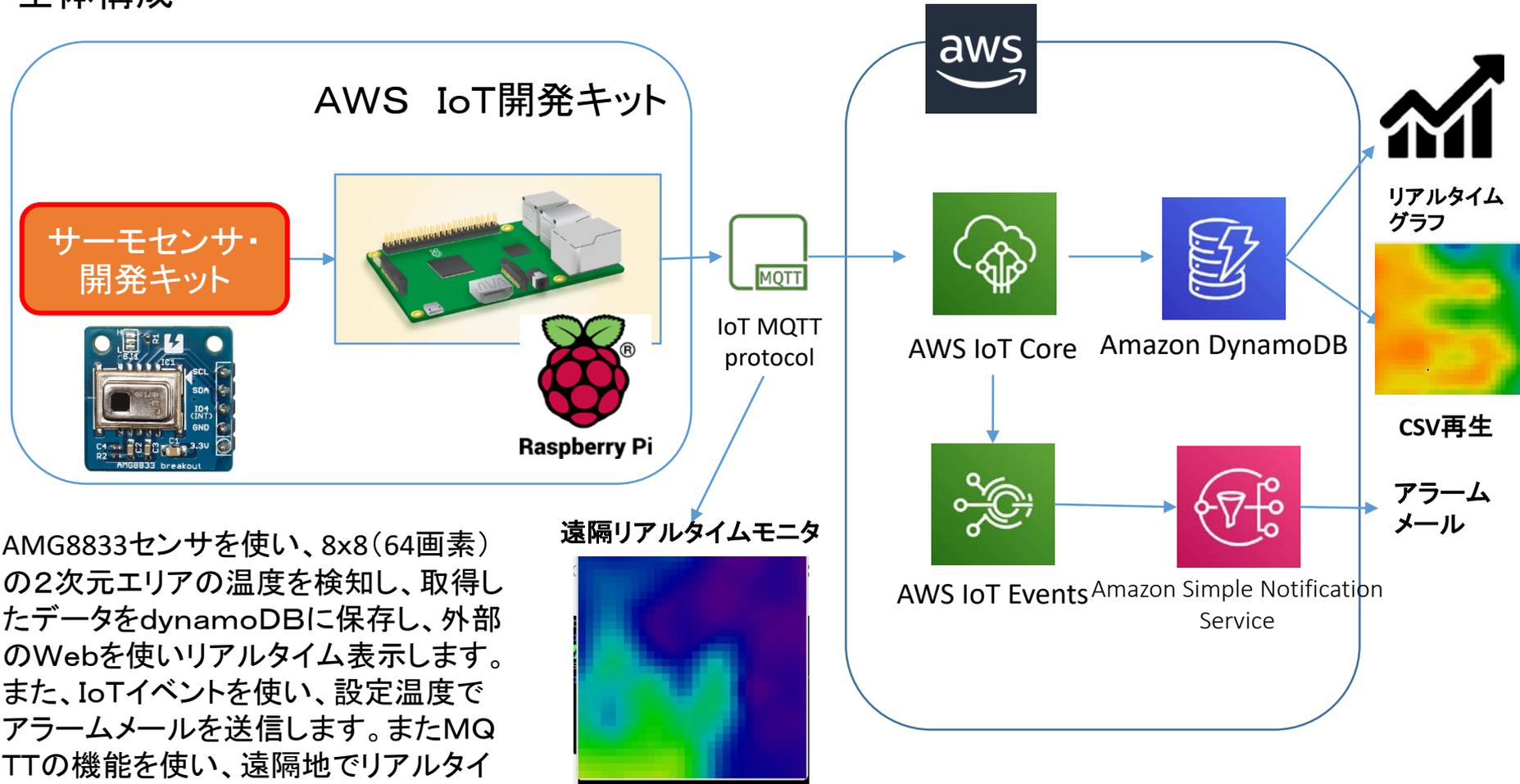
The screenshot shows the AWS IoT console interface. On the left sidebar, the '安全性' (Security) menu is expanded, and '証明書' (Certificates) is highlighted with a red box. The main content area displays a list of certificates under the heading '証明書 ポリシー作成3'. A context menu is open over one of the certificate cards, with the option 'ポリシーのアタッチ' (Attach Policy) highlighted with a red box. A red text overlay on the right side of the screenshot reads: 'モノの登録で作成した証明書にポリシーをアタッチします。右クリック > ポリシーのアタッチ'.



AWS IoT開発

④サーモセンサ情報収集、アラーム送付

全体構成



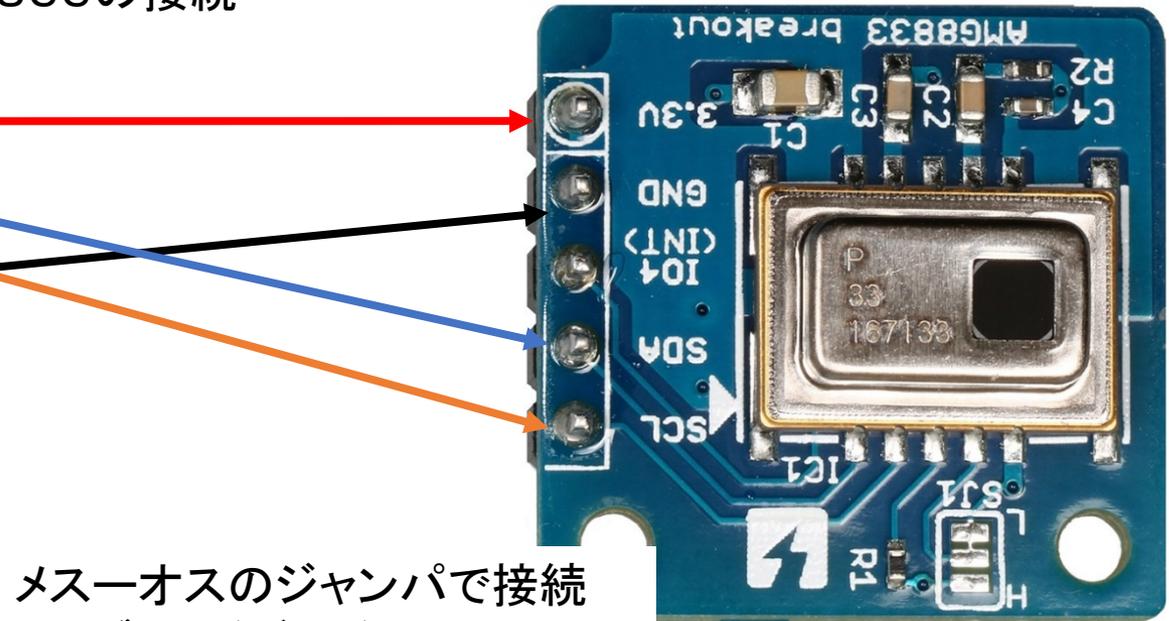
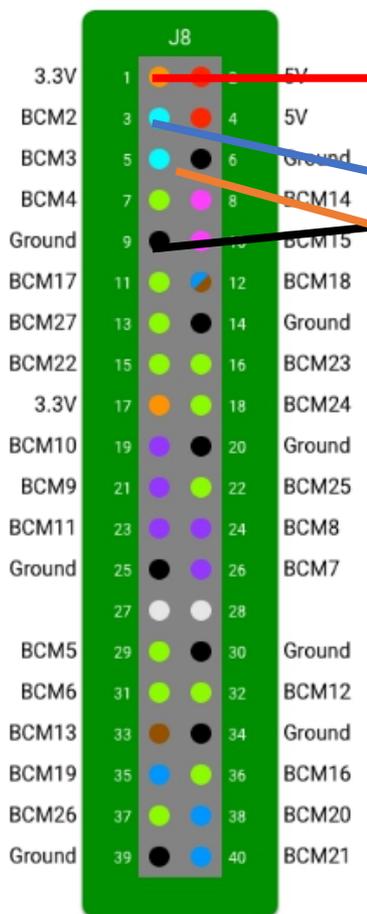
AMG8833センサを使い、8x8(64画素)の2次元エリアの温度を検知し、取得したデータをdynamoDBに保存し、外部のWebを使いリアルタイム表示します。また、IoTイベントを使い、設定温度でアラームメールを送信します。またMQTTの機能を使い、遠隔地でリアルタイムのモニタも可能

AWS IoT開発

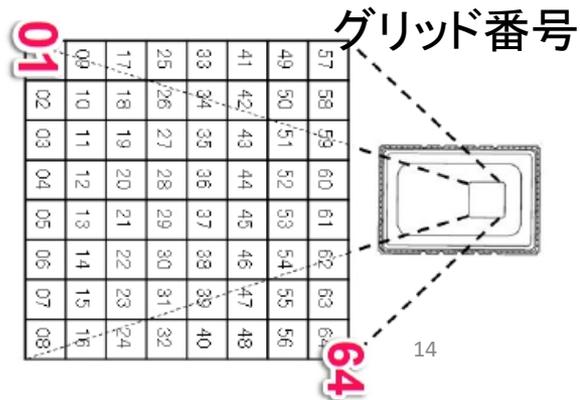
④サーモセンサ情報収集、アラーム送付

AWS IoT Core

Raspberry PiとAMG8833の接続



メスオスのジャンプで接続
 Pi ブレッドボード AMG8833
 1-----+ VDD
 3-----SDA
 5-----SCL
 9-----GND





AWS IoT開発

④サーモセンサ情報収集、アラーム

AMG8833単体試験

- Piコンソールでpython pixels_test.pyを動作させます。
- 数字の羅列が出力されます。
- 出力されない場合は、I2Cの設定を確認。0x68に値が表示されれば正常。

```
$ i2cdetect -y 1
```

```
# cd /home/pi/Documents/Adafruit_AMG88xx_python/examples # python python pixels_test.py
```

```
pi@raspberrypi: ~/Documents/A...ruit_AMG88xx_python/examples
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
^CTraceback (most recent call last):
  File "thermal_cam.py", line 76, in <module>
    pygame.draw.rect(lcd, colors[constrain(int(pixel), 0, COLORDEPTH- 1)], (displayPixelHeight * ix, displayPixelWidth * jx, displayPixelHeight, displayPixelWidth))
KeyboardInterrupt
pi@raspberrypi:~/Documents/Adafruit_AMG88xx_python/examples $ python thermistor_test.py
Thermistor Temp = 21.94 *C
pi@raspberrypi:~/Documents/Adafruit_AMG88xx_python/examples $ python pixels_test.py
[17.75, 18.0, 17.75, 18.0, 17.75, 17.0, 18.0, 18.75, 17.5, 17.75, 17.5, 17.75, 17.75, 17.25, 17.75, 18.0, 18.25, 18.0, 17.5, 17.75, 18.0, 17.25, 17.5, 18.5, 18.25, 17.75, 17.75, 17.5, 17.75, 17.25, 18.0, 18.0, 17.5, 18.0, 17.25, 17.75, 17.5, 17.75, 17.75, 18.25, 17.5, 17.75, 17.75, 16.75, 17.5, 17.25, 17.75, 18.25, 18.0, 18.25, 17.5, 17.5, 17.0, 17.5, 17.0, 16.75, 18.25, 17.25, 16.75, 16.5, 17.0, 18.0, 17.0]
[17.5, 17.75, 18.0, 18.0, 18.5, 18.0, 18.0, 17.75, 17.5, 18.0, 17.75, 17.25, 17.75, 17.75, 18.0, 18.0, 18.0, 17.75, 18.0, 17.5, 17.75, 17.75, 17.75, 17.75, 17.5, 17.25, 18.0, 18.0, 17.25, 17.0, 18.0, 17.5, 18.25, 17.75, 17.5, 18.0, 17.5, 17.75, 17.0, 18.25, 18.0, 17.75, 18.0, 17.75, 17.25, 17.75, 17.75, 17.75, 17.75, 18.5, 17.5, 17.25, 16.75, 18.0, 18.0, 16.75, 18.5, 17.5, 17.0, 16.5, 17.0, 17.0, 17.25]
[18.0, 17.75, 17.75, 18.25, 18.0, 17.5, 18.25, 18.25, 17.75, 18.25, 17.75, 17.5,
```



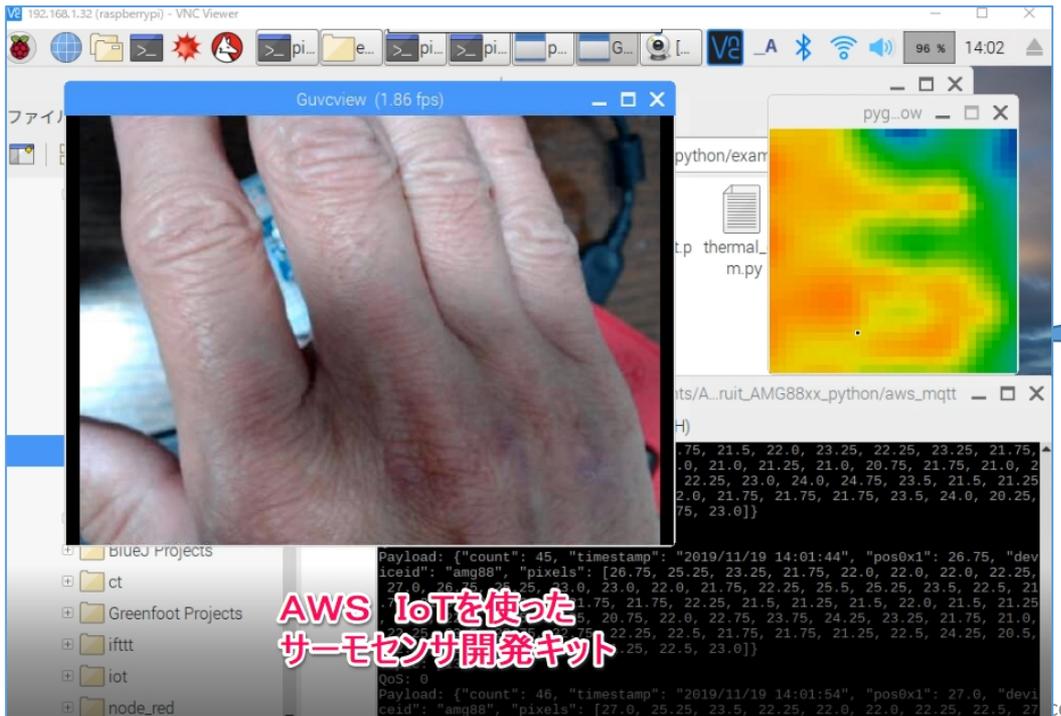
AWS IoT開発

④サーモセンサ情報収集、アラーム

AMG8833 画像試験

- Piコンソールでpython thermal_cam.pyを動作させます。
- 240x240ピクセルの画面で表示されます。
- プログラムで、最低:20度、最高:32度に設定しています。適宜数字を変えてください
- 現在の温度は、python thermistor_test.pyでテストしてください。

```
# cd /home/pi/Documents/Adafruit_AMG88xx_python/examples # python thermal_cam.py
```



手をかざした状態



AWS IoT開発

④サーモセンサ情報収集、アラーム送付

AWS IoT折り返し試験

- Piのコマンド画面からsubscriber, publisherのやりとりを実施
- subscribe側に温度のデータ(リスト形式;64個)が受信できていればOk, なおテスト用にpos0x1の最初のデータをグラフ表示のため取り出し

The screenshot shows a VNC viewer of a Raspberry Pi. The main terminal window displays MQTT messages received from a publisher. The messages contain a list of 64 temperature values (pixels) and a timestamp. A secondary terminal window shows a heatmap visualization of the temperature data, with a color scale from blue (low) to red (high). The terminal output includes the following JSON payload:

```

Payload: {"count": 29, "timestamp": "2019/11/16 19:58:28", "pos0x0": 22.75, "deviceid": "amg88", "pixels": [22.75, 22.5, 21.75, 22.0, 24.0, 23.75, 25.0, 26.75, 22.5, 23.0, 21.25, 22.25, 22.75, 23.25, 27.0, 28.5, 22.75, 22.0, 22.0, 26.25, 25.75, 24.0, 26.5, 28.75, 22.5, 20.5, 20.75, 22.0, 25.0, 26.75, 27.75, 28.75, 21.0, 20.5, 21.0, 22.5, 24.25, 26.25, 27.75, 28.75, 20.75, 21.0, 21.25, 25.25, 25.75, 24.75, 23.5, 22.5, 20.5, 20.5, 20.5, 21.0, 21.5, 21.25, 21.75, 22.0, 19.25, 20.0, 19.75, 20.75, 20.25, 20.75, 21.5, 21.5]}
    
```

Below the terminal output, there is a list of "Message Published..." entries. A status bar at the bottom indicates "18.6 GiB (合計: 26.6 GiB)".

```

# cd pi>Documents>Adafruit_AMG88xx_python>aws_mqtt
# python mqtt_subscribe_json.py 受信側を先に立ち上げ
別のウインドを開き
# python amg88_mqtt_publish_json.py 送信側立ち上げ
    
```



AWS IoT開発

④サーモセンサ情報収集、アラーム送付

リアルタイム表示

- IoT. Kyoto様の利用>グラフ
- Deviceid: amg88 (pythonで書き込み済)
- 表示対象をチェックすると自動で表示されます。



8x8=64個のデータのうち、0x0のデータのみ表示。他はリスト形式でDBに確保



AWS IoT開発

④サーモセンサ情報収集、アラーム送付

IoTイベント設定

- 指定設定温度で、メール送信するため、IoTイベントを設定します。
- 探知器モデル > 探知器モデルの作成
- 新しく作成する

aws サービス リソースグループ spectrum-tech オハイオ サポート

IoT イベント

探知器モデル

探知器モデル

アクション 探知器モデルの作成

| 名前 | 作成日 | 説明 |
|--|-----------------|--|
| detecthumanmodel | Sun Nov 17 2019 | |
| AWS_IoTEvents_Hello_World1573622847581 | Wed Nov 13 2019 | A sample "Hello_World" detector model that monitors powerwall voltage. |

aws サービス リソースグループ

タイトルなし 編集

+

AWS IoT イベント

台方法

るには、デバイスの状態 探知器モデルを作成しま

作成する

新しいディテクターモデルを作成

新しく作成する

テンプレート

業界固有のレートを選択

テンプレートし、洗濯

探知器モデルでの入力の使用

入力、探知器モデルで監視して使用するテレメトリデータ(す。入力値は、アクションを起動したり状態を変更したりする入力値は、左のナビゲーションから利用できる【入力】ページで

フィードバック 日本語

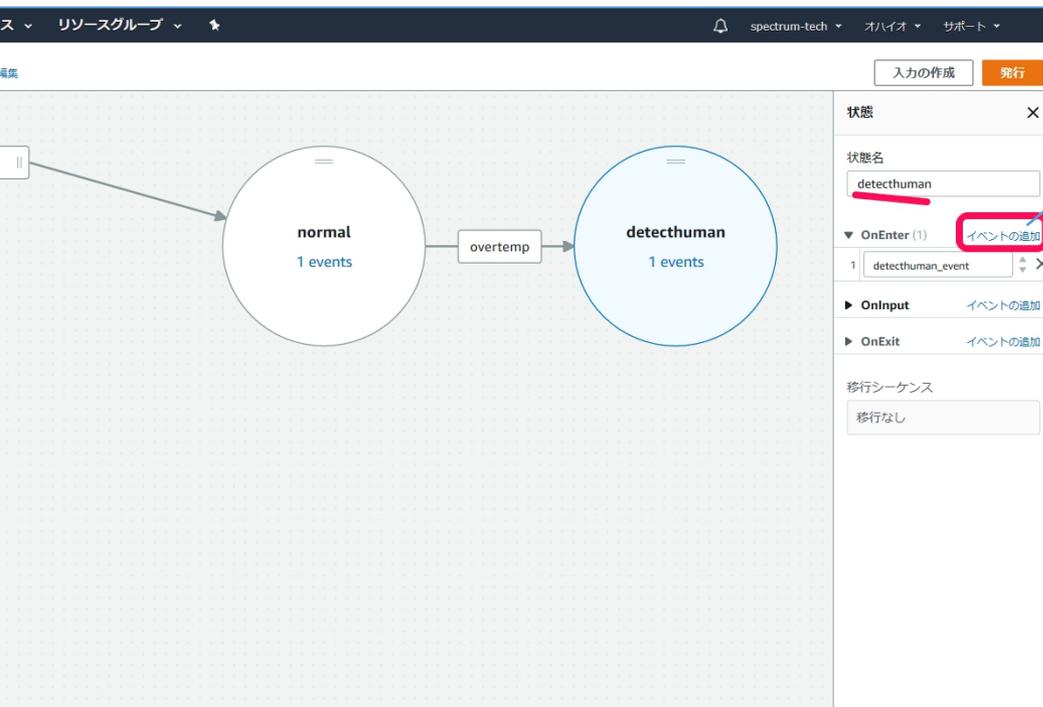
© 2008 - 2019, Amazon Web Services, Inc. またはその関連会社。無断転用禁止。 プライバシーポリシー 利用規約

AWS IoT開発

④サーモセンサ情報収集、アラーム送付

IoTイベント設定

- 状態名: detecthuman
- イベントの追加
- イベント名: detecthuman_event
- イベント条件: `$variable.tempalarm==1`
- イベントアクション: SNSメッセージの送信



OnEnter イベントの追加 ✕

イベント名

イベントの条件 - オプション
 探知器モデルがこのイベントをトリガーすることを知らせる入力およびその属性値を入力します。

イベントアクション
 イベントがトリガーされたときに実行するアクションを選択します。

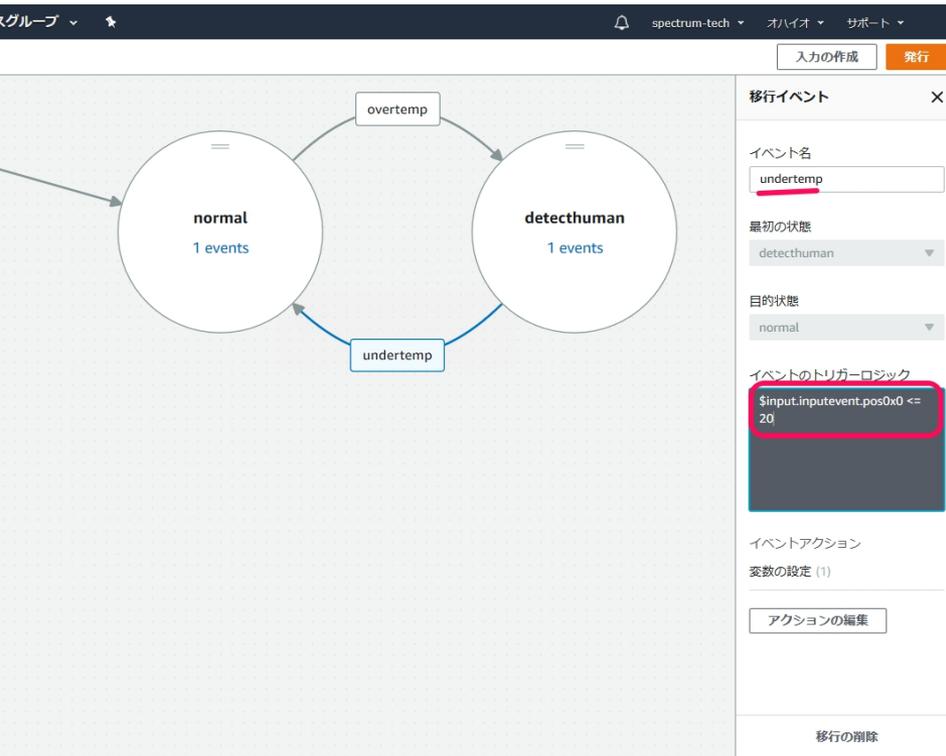
▶

AWS IoT開発

④サーモセンサ情報収集、アラーム送付

IoTイベント設定

- 移行イベントの作成
- イベント名: undertemp
- トリガーロジック: $\$input.inputevent.pos0x0 \leq 20$
- 変数の設定 > 値の割り当て
- 変数名: tempalarm 値: 0



移行イベントアクションの編集

イベントアクション
イベントがトリガーされたときに実行するアクションを選択します。

変数の設定 Remove

変数オペレーション
この変数の値が変更される方法を選択します。
値の割り当て

変数名
既存の変数の名前を指定するか、新しい変数の名前を指定して作成します。
tempalarm

値の割り当て
変数に手動で値を割り当てます。
0

Add action Cancel Save

温度が20度を超えたら tempalarmを1に

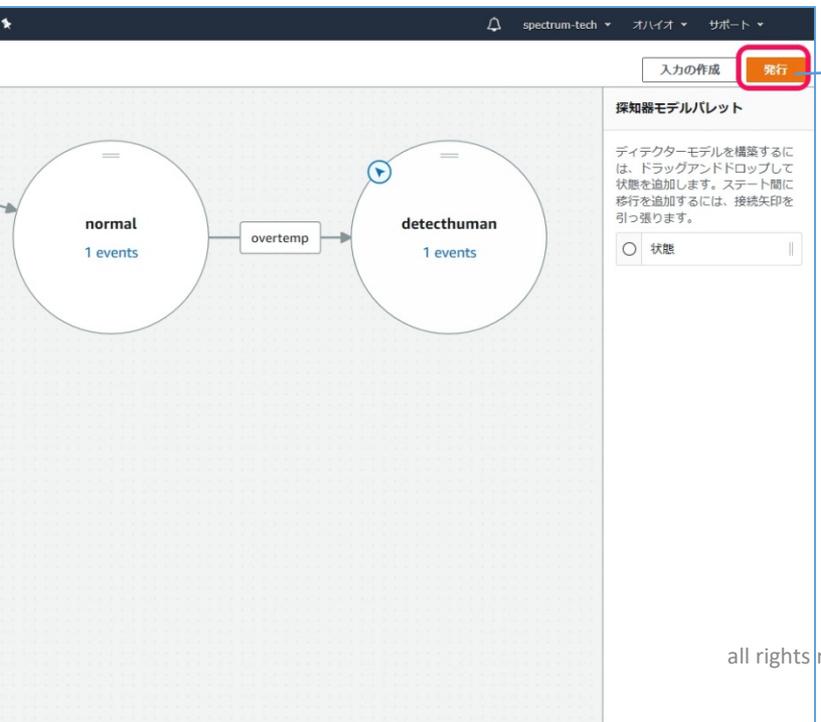


AWS IoT開発

④サーモセンサ情報収集、アラーム送付

IoTイベント設定

- 探知器モデルの発行
- モデル名 : detecthumanmodel
- ロール : myevent_role 自動設定されます。
- 単一の探知器作成、シリアル評価
- 保存して発行。エラーが出なければ作成完了



The screenshot shows the 'Publish Detector Model' dialog box. The 'Detector Model Name' is 'detecthumanmodel'. The 'Role' is 'myevent_role'. The 'Detector Generation Method' is 'Single detector creation'. The 'Detector Evaluation Method' is 'Serial evaluation'. The 'Save and Publish' button is highlighted in red.



AWS IoT開発

④サーモセンサ情報収集、アラーム送付

```
$ cd /home/pi/Documents/Adafruit_AMG88xx_python/aws_mqtt $ python mqtt_subscribe_mon.py
```

遠隔リアルタイム・モニタ

- Mqttのsubscribe機能を使って、遠隔地でリアルタイムにサーモセンサの画像のモニタを行います。Mqttの送信タイミングの10秒毎受信
- python mqtt_subscribe_mon.py

遠隔リアルタイムモニタ

サーモセンサ画像の温度は、10-32度で設定しています。プログラムで修正してください



AWS IoT開発

④サーモセンサ情報収集、アラーム送付

CSVの画像再生

- DynamoDBのCSVエクスポートしたoutput.csvを画像再生します
- python thermal_csv.py

```
$ cd /home/pi/Documents/Adafruit_AMG88xx_python/DynamoDBtoCSV $ python thermal_csv.py
```

The screenshot shows a terminal window with a heatmap visualization on the left and a CSV data stream on the right. A red text overlay reads "1秒毎に再生します。" (Regenerates every 1 second). The CSV data consists of multiple rows of numerical values, likely representing sensor readings over time. The terminal window title is "pi@raspberrypi: ~/Documents/A...G88xx_python/DynamoDBtoCSV".