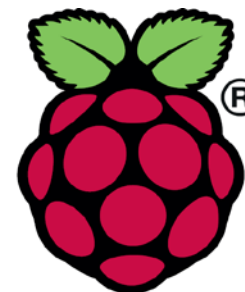


# はじめてのAWS IoT開発キット ～センサ、RaspberryPiを使い、実体験。開発短縮～ 実践編（抜粋版）



**AWS IoT**



**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](mailto:sales@spectrum-tech.co.jp)

# AWS IoT開発キット 目次

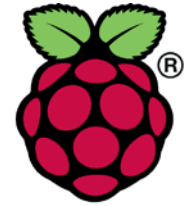
## Pi運用マニュアル

	ページ
1. RaspberryPiについて	3
2. Linux基本コマンド	4
3. 基本操作	5
4. 日常運用(ウイルススキャン、更新)	6

## AWS IoT開発

	ページ
① メニュー	8
② AWS設定	9
③ AWS IoT設定	16
④ LED点灯	29
• Raspberry PiのI/O	30
• ブレッドボードの使い方	31
⑤ メール送信	41
• Amazon SNS設定	43
• AWS IoTルール設定	46
• IAM確認	51
⑥ センサ情報リアルタイム表示	55
• DynamoDB確認	66
• IAMユーザ追加	69
• Cloudwatch使い方	74

抜粋の  
ため  
ページ  
がずれ  
てます



## Pi運用マニュアル

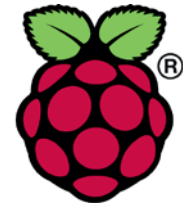
### 1. Raspberry Piについて

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



Raspberry Pi

# 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

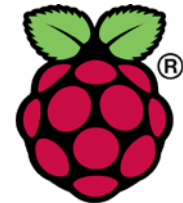
root@:~ \$ **su -**      スーパーユーザ(root)に切り替え、パスワードを入力  
 root@:~# **ps a**      現状の動いているプロセスを表示  
 root@:~# **kill**      特定のプロセスを強制終了

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

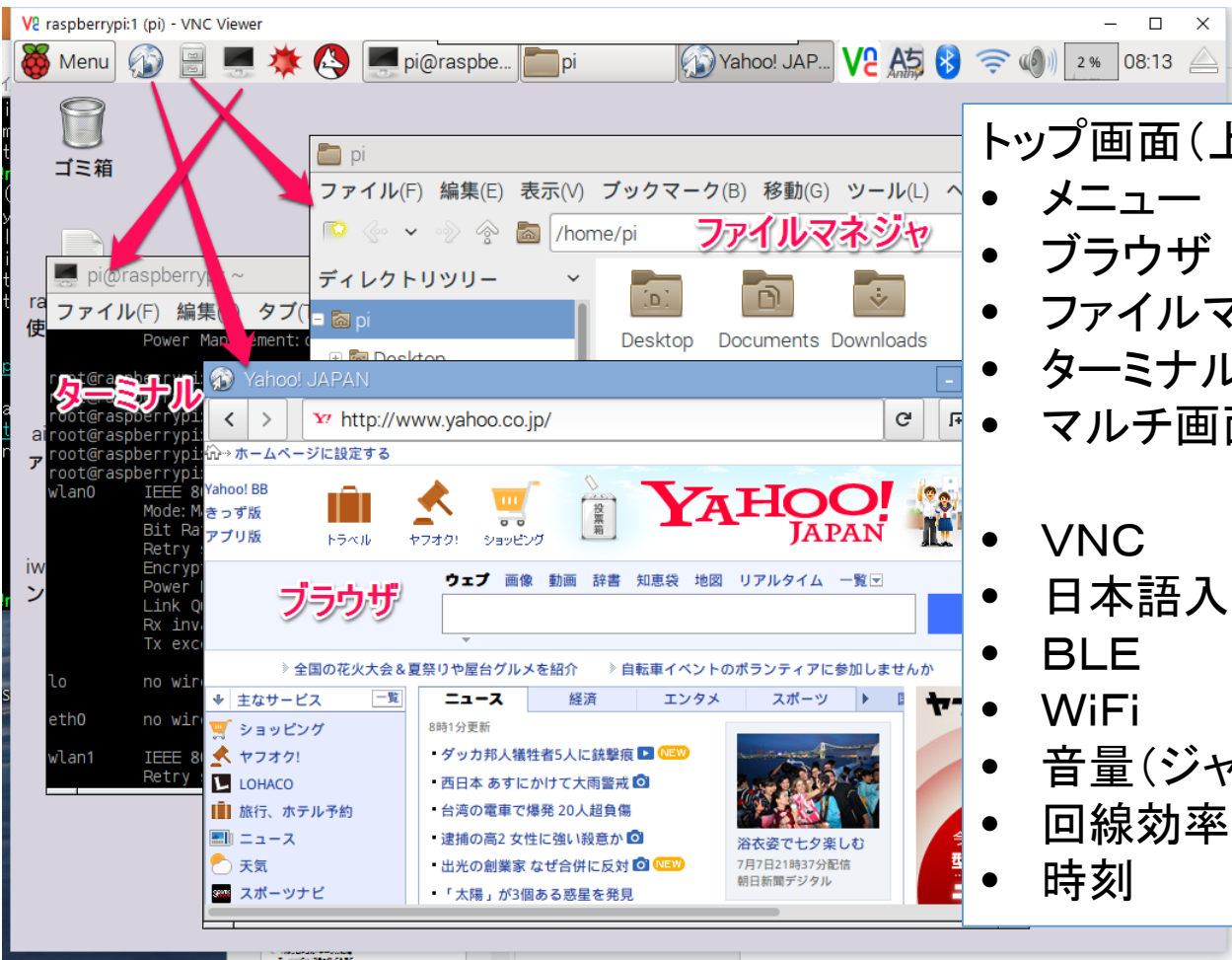


Raspberry Pi

## Pi運用マニュアル

### 3. 基本操作

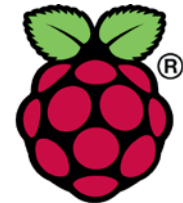
#### ① 表示画面と内容



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
- WiFi
- 音量(ジャックで聴けます)
- 回線効率
- 時刻



## Pi運用マニュアル

### 4. 日常運用

- ① セキュリティ対策(アンチウイルス更新、スキャン)
  - アンチウイルス対策として無料のclamAVをインストールしてます。
  - 手動での運用を基本としています。

```
pi@raspberrypi: ~
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
ERROR: /var/log/clamav/freshclam.log is locked by another
ERROR: Problem with internal logger (UpdateLogFile = /var/
og).
root@raspberrypi: ~# leafpad /etc/clamav/freshclam.conf
root@raspberrypi: ~# freshclam
ClamAV update process started at Fri J
main.cvd is up to date (version: 57, sigs: 4216790, f-level: 60, builder: mishh
ammer)
daily.cvd is up to date (version: 21862, sigs: 394456, f-level: 63, builder: neo
)
bytecode.cvd is up to date (version: 283, sigs: 53, f-level: 63, builder: neo)
root@raspberrypi: ~# clamscan --infected --remove --recursive

SCAN SUMMARY
Known viruses: 4607906
Engine version: 0.99.2
Scanned directories: 264
Scanned files: 2063
Infected files: 0
Data scanned: 61.31 MB
Data read: 49.02 MB (ratio 1.25:1)
Time: 71.844 sec (1 m 11 s)
root@raspberrypi: ~#
```

パターンファイル更新  
手動スキャン時に更新されます  
手動でスキャン  
# clamscan --infected --remove --recursive  
自動化可能ですが、バックグラウンドで重くなる可能性大。コマンド入力後約25分位かかります。

手動でスキャン







## AWS IoT開発

### ①. メニュー

- AWS設定

AWS IoTを実施するためのAWSの設定  
AWSの詳細説明は以下のURLへ

[https://aws.amazon.com/jp/cloud/?nc2=h\\_l2\\_cc](https://aws.amazon.com/jp/cloud/?nc2=h_l2_cc)

- AWS IoT設定

AWS IoTのデバイス、証明書、ポリシー、ルールなどの設定を行います。  
開発者ガイドは以下のURLへ

[https://docs.aws.amazon.com/ja\\_jp/iot/latest/developerguide/what-is-aws-iot.html](https://docs.aws.amazon.com/ja_jp/iot/latest/developerguide/what-is-aws-iot.html)

- LED点灯

AWS IoTのshadowを使って、Raspberry Piに接続したLEDを遠隔から点灯させます

- メール送信

タクトスイッチを使い、MQTTプロトコルを介して、AWS IoTに接続し、AWS SNSからE-mailの送信を行います。上記同様単体試験などのプログラムを提供

- センサ情報リアルタイム表示

BMP280センサを使いMQTTプロトコルを介して、AWS IoTに接続し、AWS DynamoDBにデータを保存し、そのデータを外部ツールでリアルタイム表示します。またその過程の、Raspberry Piでのセンサ単体試験、AWS IoT折り返し試験などのプログラムも提供。順序だてて構築して行きます。



## 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

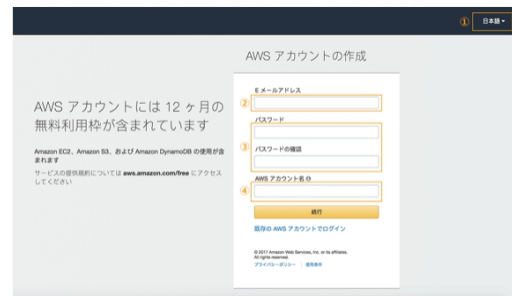
AWS アカウントを作成すると、1年間の無料利用枠はもちろん、AWS クラウドの世界中のリージョンで提供されるすべてのサービスを始めることができます。こちらでは日本のお客様にAWS アカウント作成におけるポイントをご紹介します。

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



※クリックすると大きな画像でご覧いただけます。

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。

※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

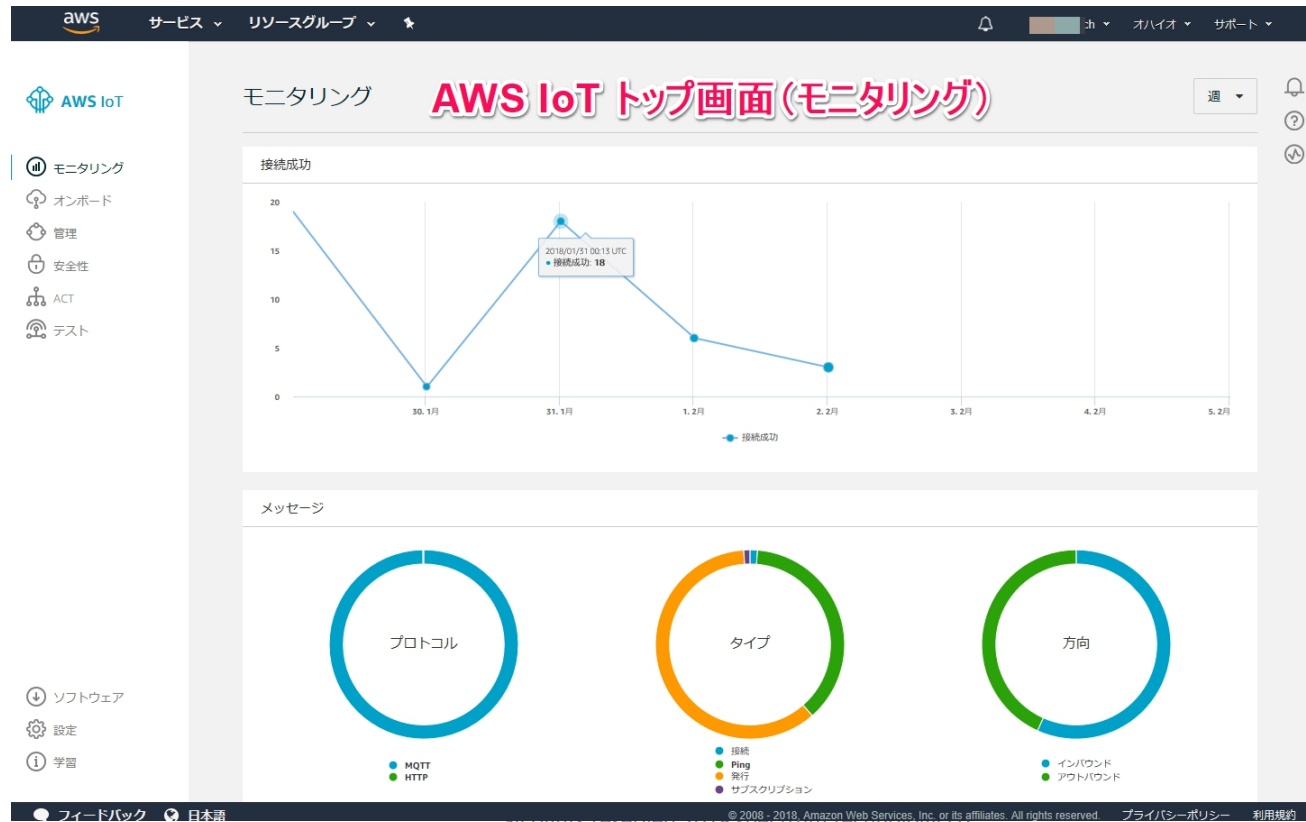


## AWS IoT開発

### 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。

今すぐ始める

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。

ボタンの設定

ボタンをお持ちでない場合 購入する

**モノ**

basicpi BASIC pi1 タイプなし

**AWS IoTに接続するモノを登録します。**

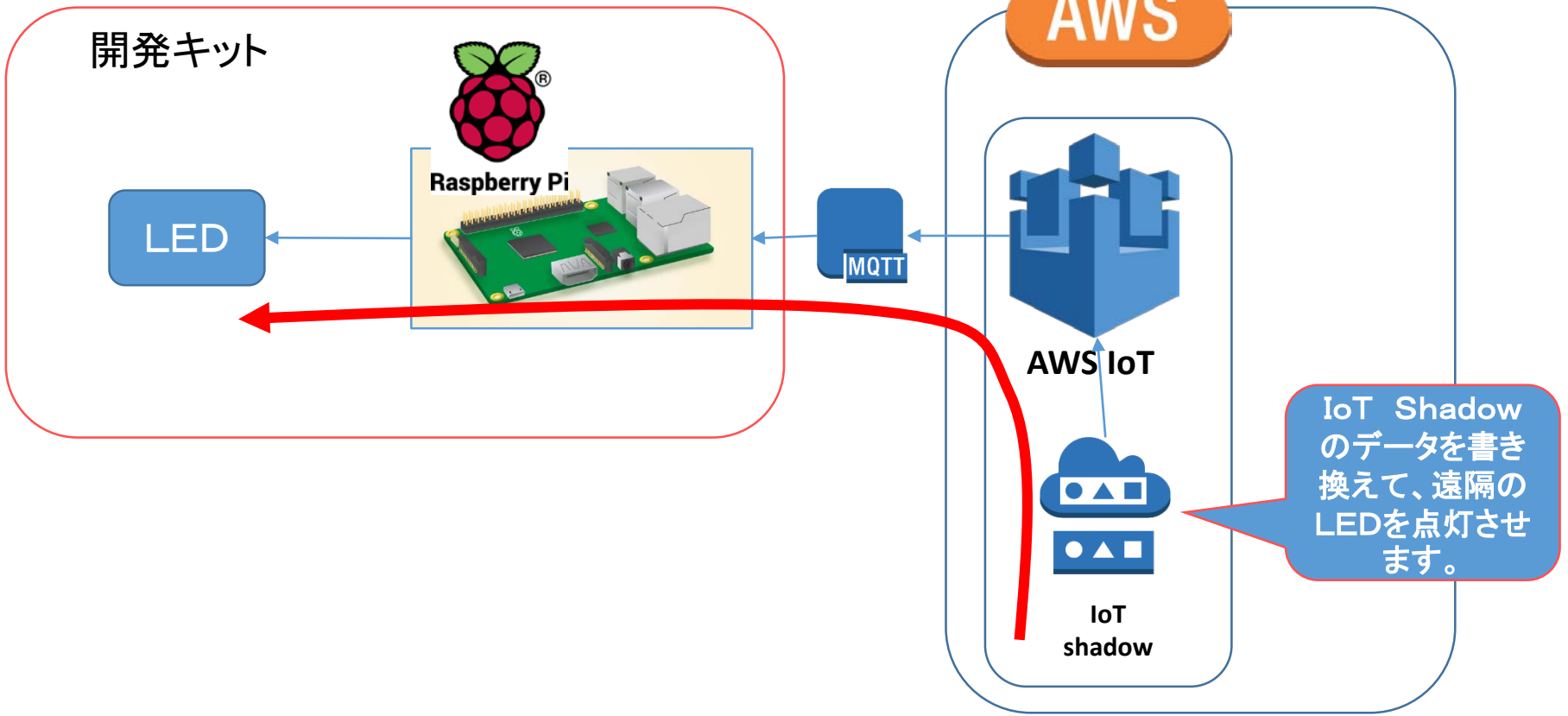
**モノに証明書、ポリシーを割り付けます。**

### ④. LED点灯



AWS IoT

全体構成



### ④. LED点灯



AWS IoT

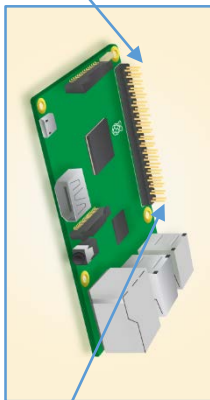
#### RaspberryPiのI/O



Raspberry Pi

- = 5V
- = 1.8V
- = GPIO
- = I2C
- = SPI
- = 3.3V
- = Ground
- = PWM
- = I2S
- = UART

J8	
3.3V	1
BCM2	3
BCM3	5
BCM4	7
Ground	9
BCM17	11
BCM27	13
BCM22	15
3.3V	17
BCM10	19
BCM9	21
BCM11	23
Ground	25
BCM5	29
BCM6	31
BCM13	33
BCM19	35
BCM26	37
Ground	39
5V	2
5V	4
Ground	6
BCM14	8
BCM15	10
BCM18	12
Ground	14
BCM23	16
BCM24	18
Ground	20
BCM25	22
BCM8	24
BCM7	26
Ground	28
BCM12	32
Ground	34
BCM16	36
BCM20	38
BCM21	40



GPIO Signal	Alternate Functions	
BCM2	I2C1 (SDA)	
BCM3	I2C1 (SCL)	
BCM7	SPI0 (SS1)	
BCM8	SPI0 (SS0)	
BCM9	SPI0 (MISO)	
BCM10	SPI0 (MOSI)	
BCM11	SPI0 (SCLK)	
BCM13	PWM1	
BCM14	UART0 (TXD)	MINIUART (TXD)
BCM15	UART0 (RXD)	MINIUART (RXD)
BCM18	I2S1 (BCLK)	PWM0
BCM19	I2S1 (LRCLK)	
BCM20	I2S1 (SDIN)	
BCM21	I2S1 (SDOUT)	

- 今回使用するピンは、9, 11番を使ってLED接続します。
- 3.3V, 5V接続時は注意しましょう。

#### Raspberry PiのI/O一覧

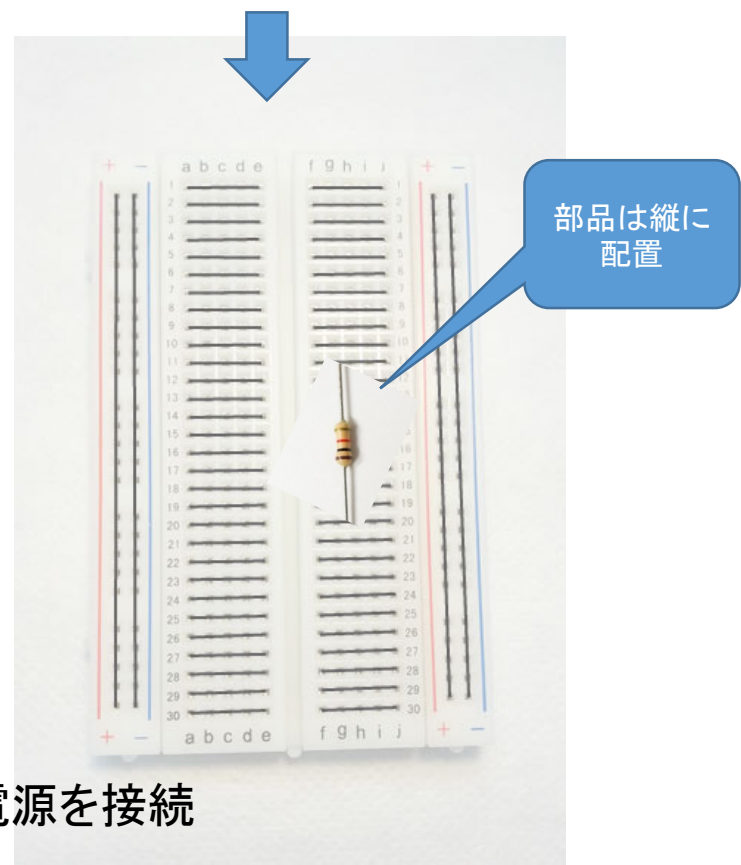
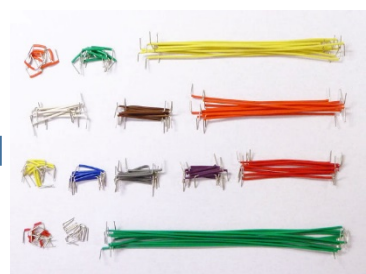
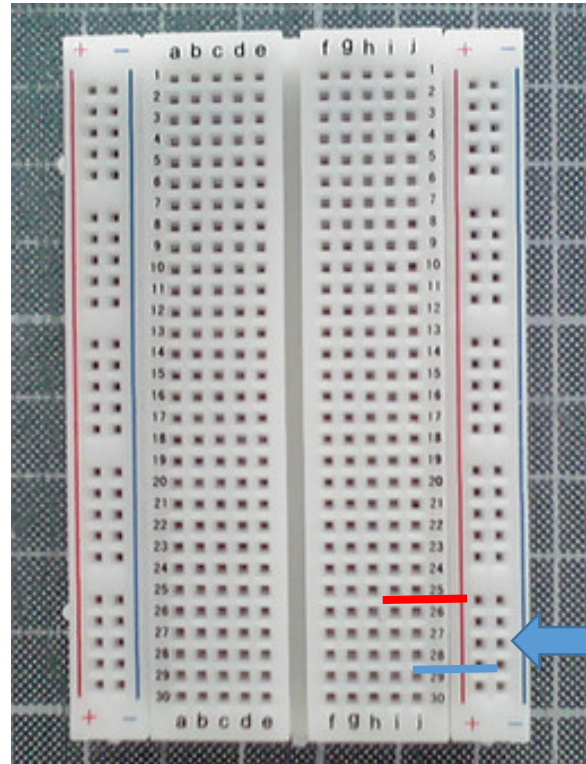
### ④. LED点灯



AWS IoT

ブレッドボードの使い方

- 電源の+、-は縦に接続されている。
- 1-30のラインは横に接続されている。



ジャンプワイヤを使って電源を接続





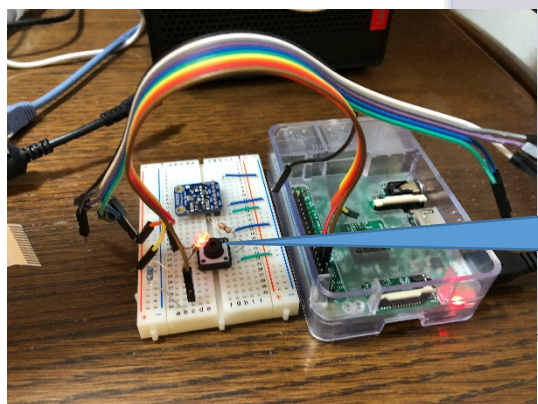
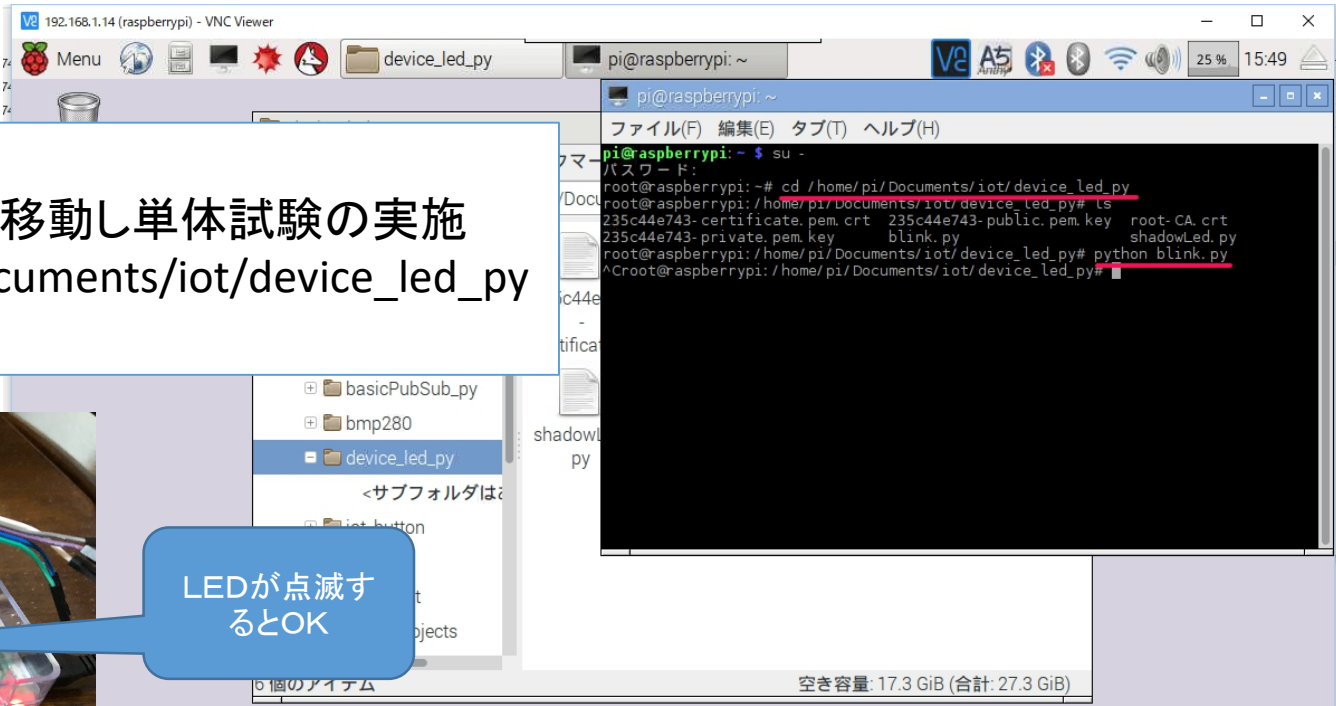
## AWS IoT開発

### ④. LED点灯

#### 単体試験

- Raspberry PiとLEDの接続を確認します。
- Raspberry piで以下のプログラム

rootで立ち上げ  
以下のフォルダに移動し単体試験の実施  
# cd /home/pi/Documents/iot/device\_led\_py  
# python blink.py



LEDが点滅するとOK





## AWS IoT開発

### ④. LED点灯

#### LED shadow試験

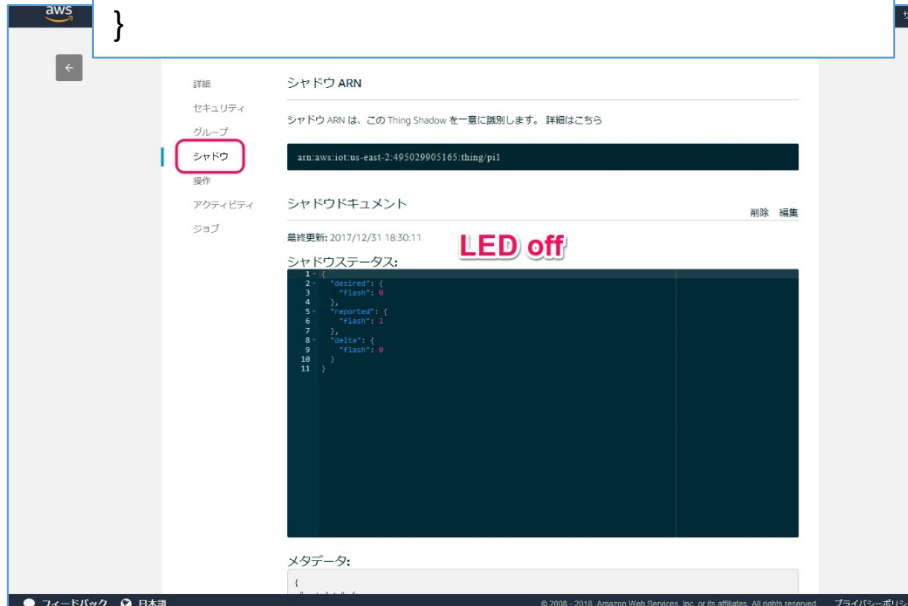
- Raspberry PiとLEDの接続を確認します。
- AWS IoTのコンソールからShadowコマンド送信しLEDランプを点灯させる。
- 管理>モノ(該当のモノ)>シャドウ

最初はLED offです。

LED onにします。

```
{ "desired": { "flash": 0 },
  "reported": { "flash": 1 },
  "delta": { "flash": 0 }
}
```

```
{ "desired": { "flash": 1 },
  "reported": { "flash": 0 },
  "delta": { "flash": 0 }
}
```





## AWS IoT開発

### ④. LED点灯

#### LED shadow試験

- Raspberry PiとLEDの接続を確認します。
- Raspberry piで以下のプログラム

rootで立ち上げ  
以下のフォルダに移動し単体試験の実施  
# cd /home/pi/Documents/iot/device\_led\_py  
# python shadowLed.py  
待ち状態になります。

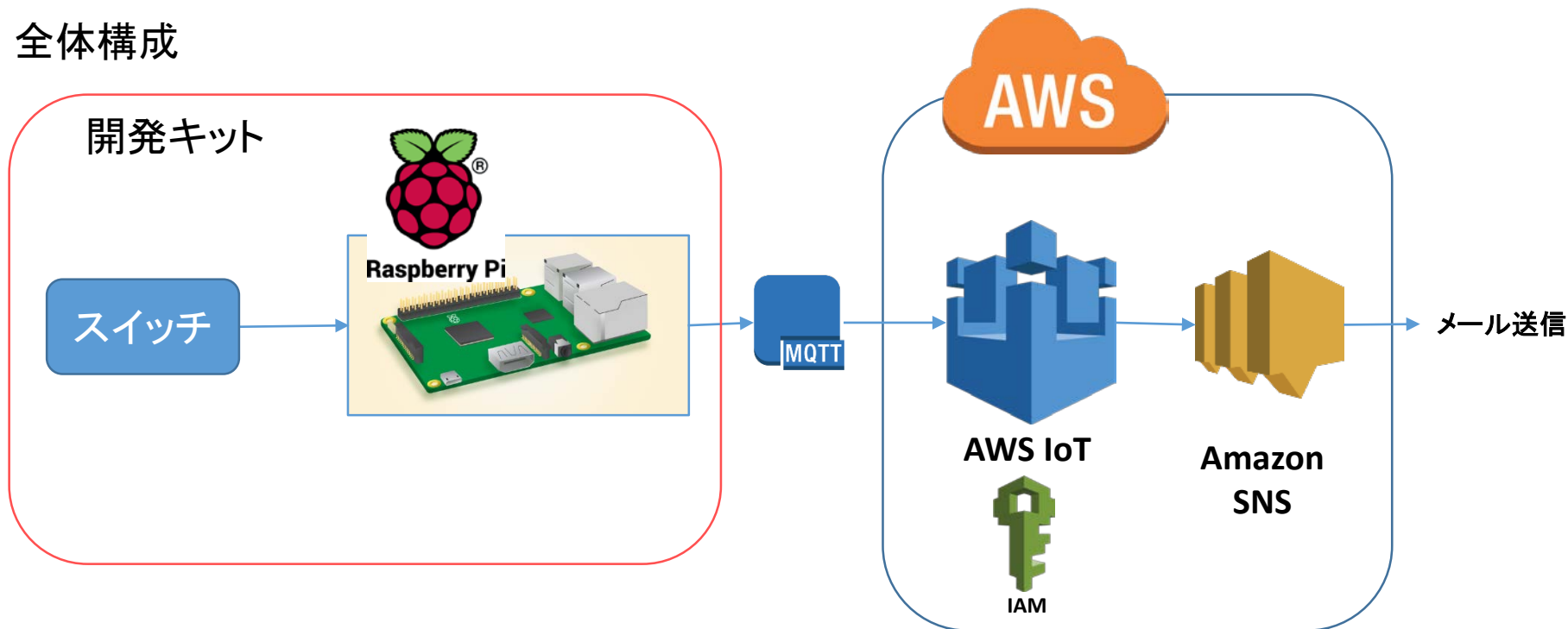
```
pi@raspberrypi: ~  
(E) タブ(T) ヘルプ(H)  
22,799 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG  
m event callback...  
22,800 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG  
ent callback is for pub/sub/unsub, removing it after invocation  
24,834 - AWSIoTPythonSDK.core.shadow.deviceShadow - INFO - Subs  
opi LED ON shadow: pi1  
42,610 - AWSIoTPythonSDK.core.protocol.connection.cores - DEBUG  
on: Resetting the backoff time to: 1 sec.  
54,862 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG  
tagel event  
018-02-05 16:07:54,878 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG  
- Dispatching [message] event  
018-02-05 16:07:56,078 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG  
- Invoking custom event callback...  
del tu/pi:1  
+++++DELTA+++++  
'timestamp': 1517814474, 'u' state': {'u' flash': 1}, 'u' version': 20, 'u' metadata':  
'u' flash': {'u' timestamp': 1517814474}}  
roperty: 1  
+++++  
DN
```

LEDランプが点灯するとともに  
Pi側の画面にONが表示されます



AWS IoT

#### 全体構成



スイッチを押すと、AWS IoTを介して、電子メールを送信します。  
AWS IoTボタンと同様の動きをシュミレーションします。





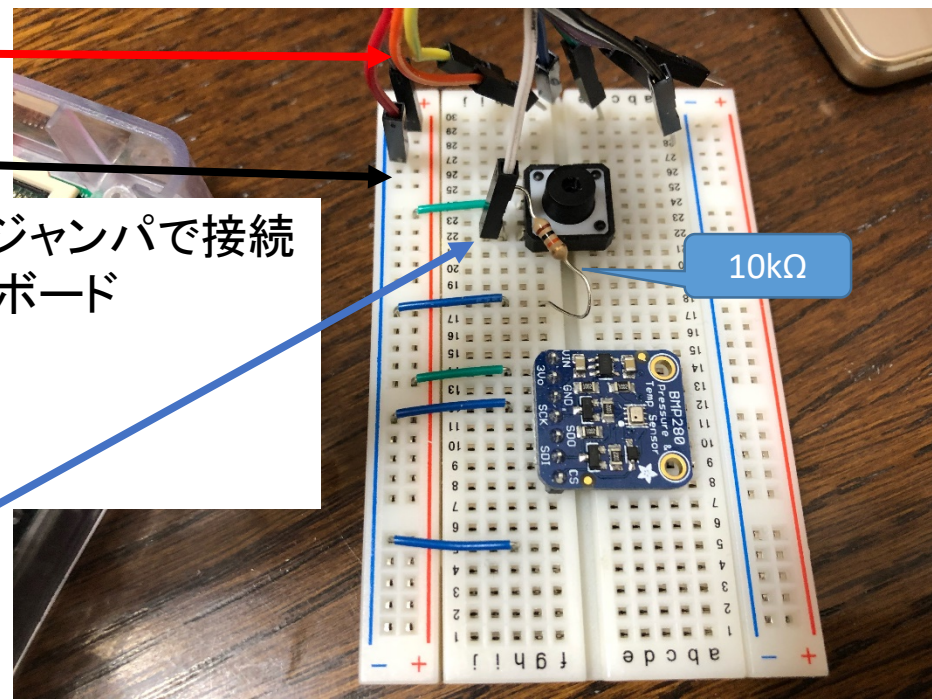
## AWS IoT開発

### ⑤. メール送信

Raspberry Piとスイッチの接続

J8			
3.3V	1	2	3
BCM2	3	4	5
BCM3	5	6	7
BCM4	7	8	9
Ground	9	10	11
BCM17	11	12	13
BCM27	13	14	15
BCM22	15	16	17
3.3V	17	18	19
BCM10	19	20	21
BCM9	21	22	23
BCM11	23	24	25
Ground	25	26	27
BCM5	29	30	31
BCM6	31	32	33
BCM13	33	34	35
BCM19	35	36	37
BCM26	37	38	39
Ground	39	40	

メスオスのジャンパで接続  
ピン ブレッドボード  
1-----+  
6-----  
36----スイッチ



### ⑤. メール送信



AWS IoT

#### Amazon SNS設定

- AWSのトップ画面でSNSを選択
- SNS>トピック>新しいトピック作成

The screenshot shows the AWS Management Console interface. On the left, the 'AWS サービス' (AWS Services) page is visible, with 'Simple Notification Service' highlighted in a red box under the 'データベース' (Database) category. On the right, the 'トピック' (Topic) page is shown, with '新しいトピックの作成' (Create new topic) highlighted in a red box. Below this, a table lists existing topics:

名前	ARN
dynamodb	arn:aws:sns:ap-northeast-1:123456789012:dynamodb
myiot_sns	arn:aws:sns:ap-northeast-1:123456789012:myiot_sns

At the bottom of the console, the footer text reads: © 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved. プライバシーポリシー

### ⑤. メール送信

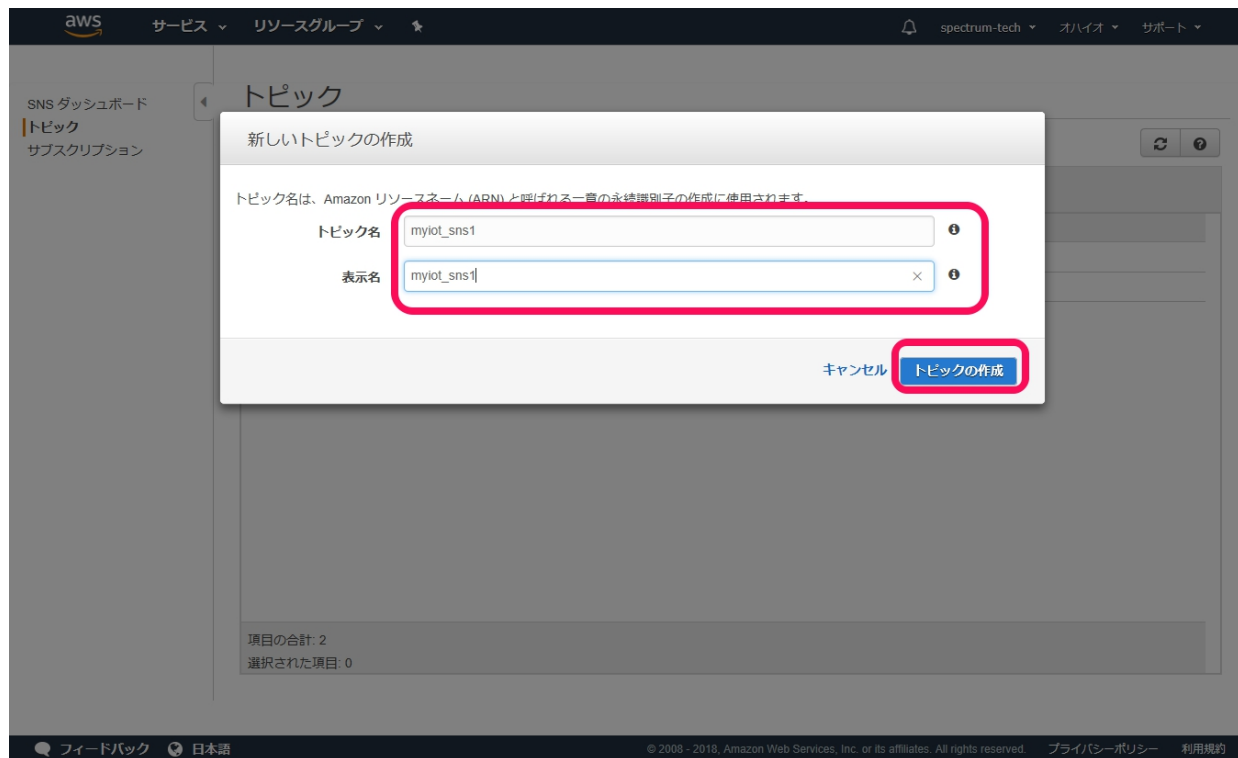


AWS IoT

#### Amazon SNS設定

- 新しいトピック作成
- トピック名
- 表示名を入力します(トピック名と同じでもOK)

例: myiot\_sns1



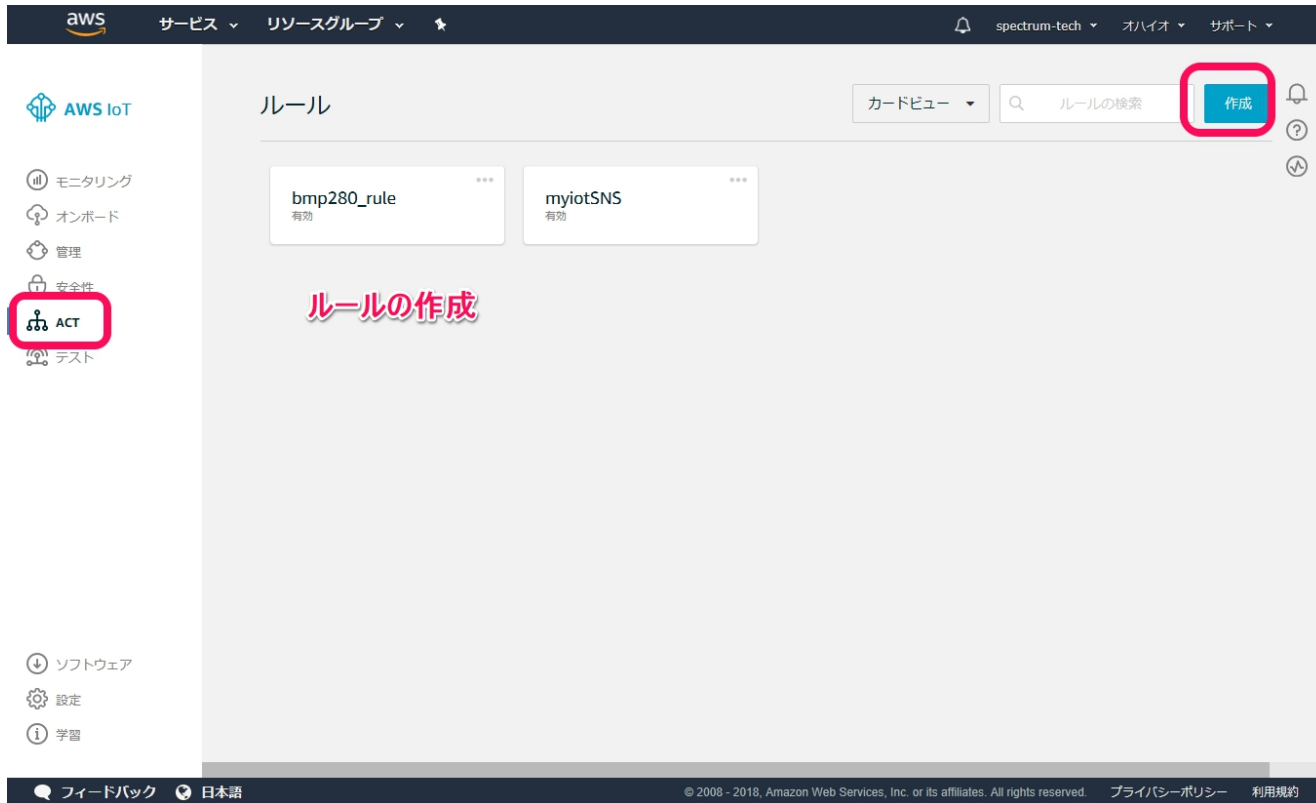
### ⑤. メール送信



AWS IoT

#### AWS IoTのルール設定

- AWS IoTのコンソールからACT>ルール
- ルール作成を押します。







## AWS IoT開発

### ⑤. メール送信

#### IAMの確認

- AWSのトップ画面からIAMを選択
- IAMで、ユーザ、ロールなどアクセス権限が一覧でわかります。

The screenshot shows the AWS IAM console interface. On the left, the navigation pane lists various services, with 'IAM' highlighted in a red box. A blue arrow points from this box to the main content area. The main content area displays the 'Identity and Access Management' overview, including sections for 'IAM リソース' (listing 2 users, 0 groups, and 17 custom policies) and 'セキュリティステータス' (listing 5 items, with 2 completed). A red text overlay at the bottom of the screenshot reads 'AWSへのアクセス権限が一覧でわかります'.

### ⑤. メール送信

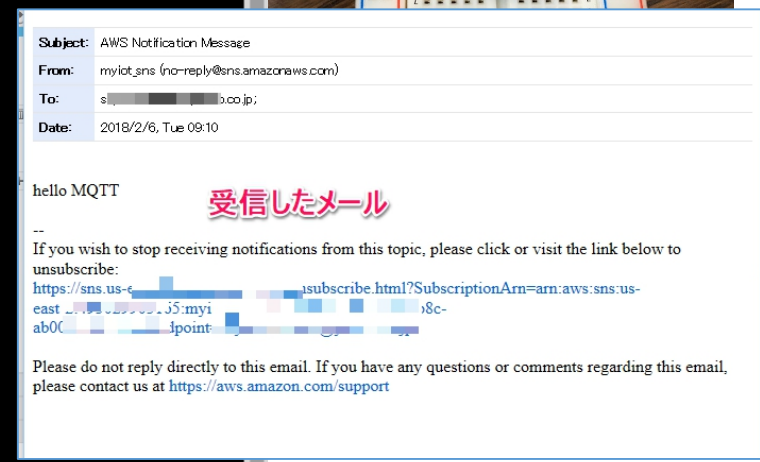
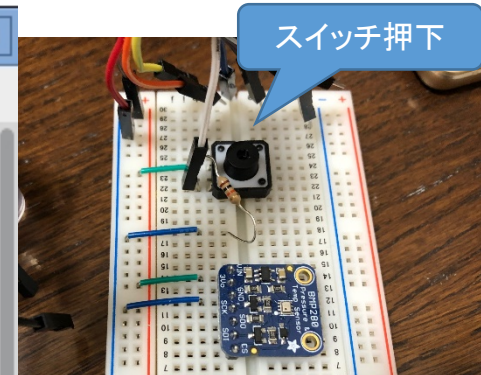
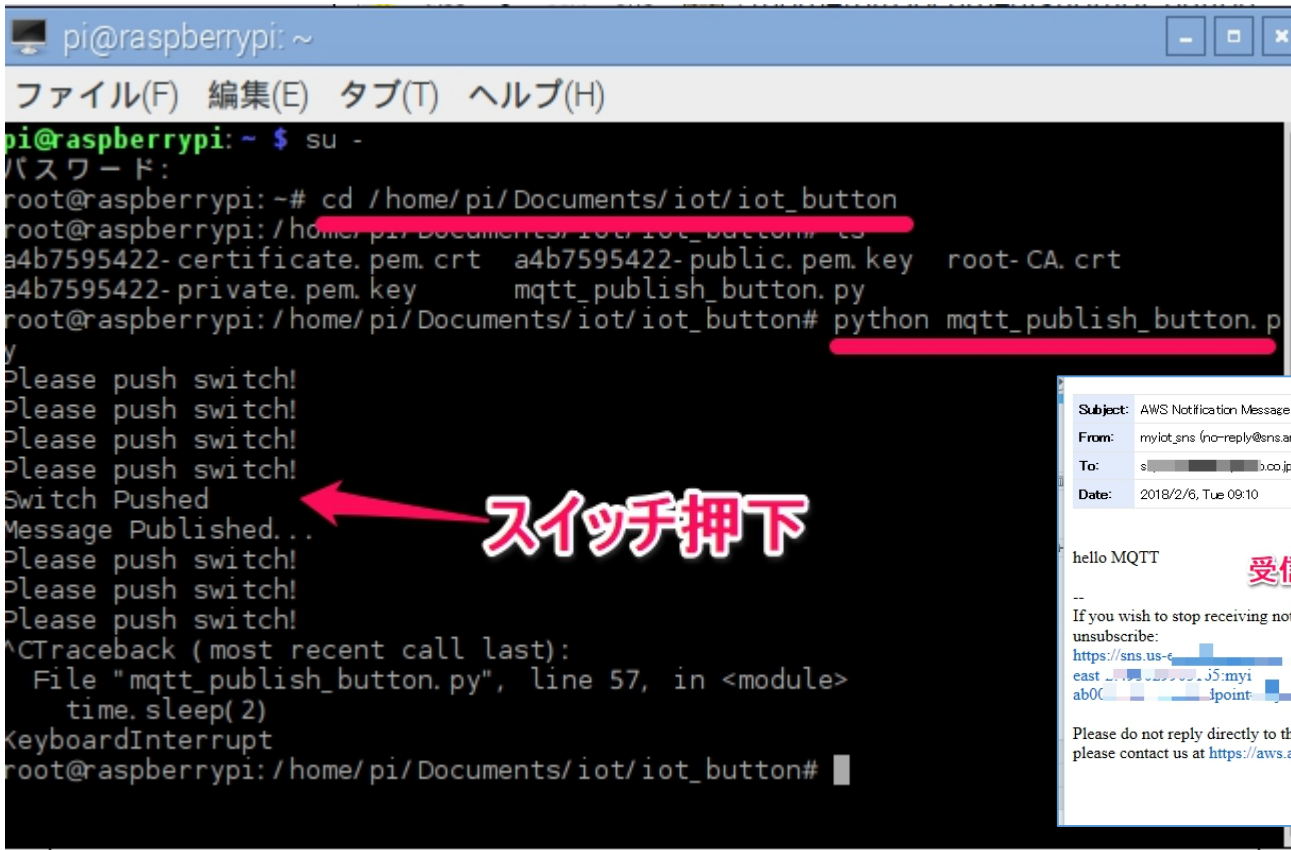


AWS IoT

#### メール送信試験

- Piからmqtt\_publish\_button.pyを動作させます。
- ブレッドボードのスイッチを押します。
- 電子メールが送信されます。

```
# cd /home/pi/Documents/iot/iot_button  
# python mqtt_publish_button.py
```



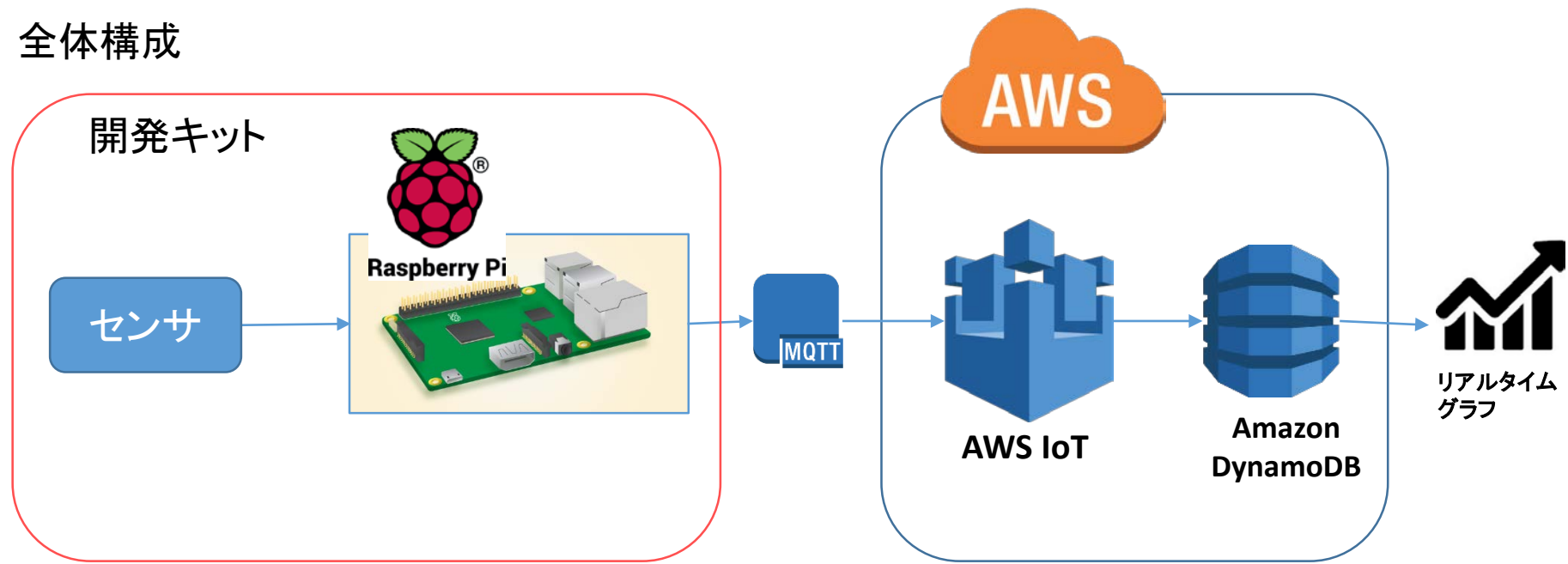


AWS IoT

## AWS IoT開発

### ⑥. センサ情報リアルタイム表示

全体構成

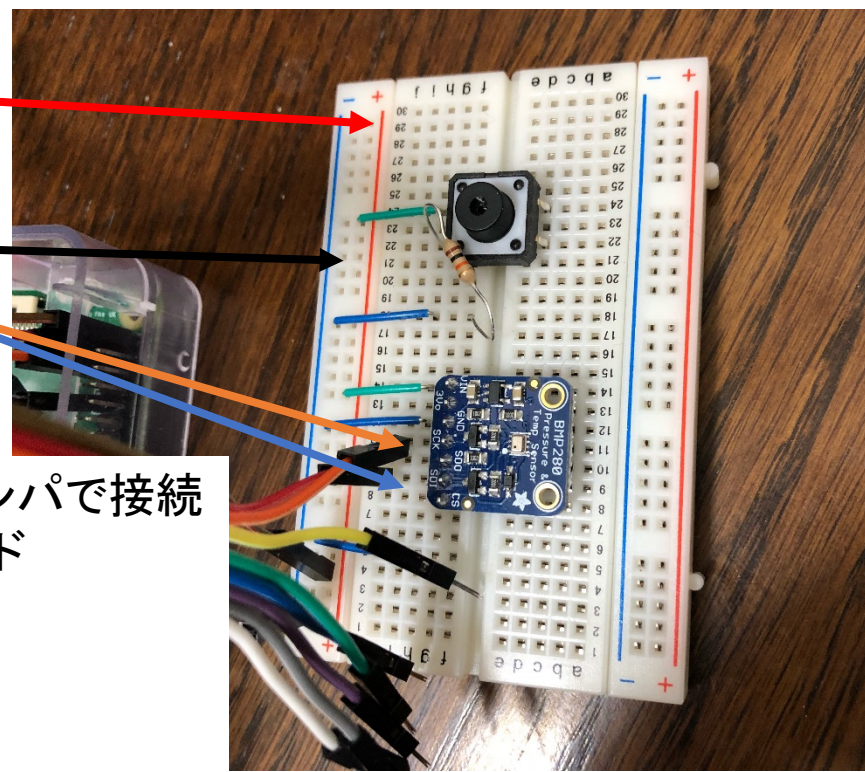
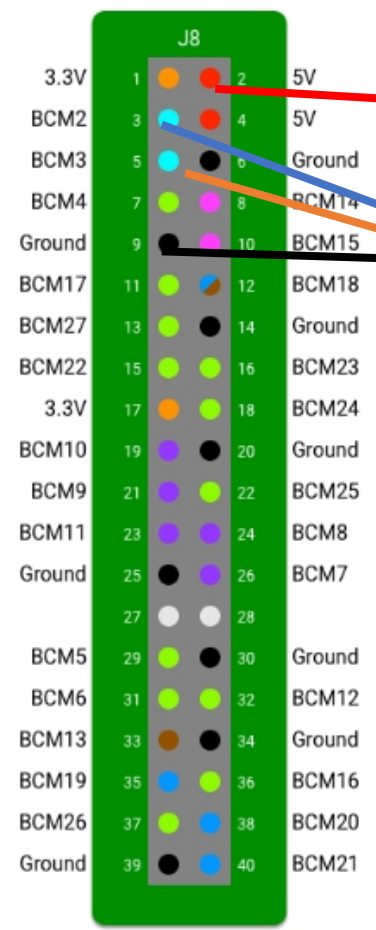


BMP280センサの気温、気圧データをAWS IoTに送信し、その内容がDynamoDBに確保されます。DynamoDBの値を外部からリアルタイムのグラフ表示します。

# AWS IoT開発

## ⑥. センサ情報リアルタイム表示

Raspberry Piとスイッチの接続



メスオスのジャンパで接続  
 ピン ブレッドボード  
 2-----+ VIN  
 3-----SDI  
 5-----SCK  
 9----- - GND



## AWS IoT開発

### ⑥. センサ情報リアルタイム表示

#### AWS IoTのルール設定

- AWS IoTのコンソールからACT>ルール
- ルール作成を押します。(メール送信と同様)

The screenshot shows the AWS IoT console interface. At the top, there's a navigation bar with 'aws', 'サービス', 'リソースグループ', and user information 'spectrum-tech'. The left sidebar contains navigation options: 'AWS IoT', 'モニタリング', 'オンボード', '管理', '安全性', 'ACT', and 'テスト'. The 'ACT' option is circled in red. The main content area is titled 'ルール' (Rules) and shows two existing rules: 'bmp280\_rule' and 'myiotSNS'. A '作成' (Create) button is circled in red in the top right of the main area. The text 'ルールの作成' (Create Rule) is overlaid in red in the center of the main area. At the bottom, there's a footer with 'フィードバック', '日本語', and copyright information.





## AWS IoT開発

### ⑥. センサ情報リアルタイム表示

#### 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側に気温、気圧のデータが受信できていればOk

The screenshot shows a VNC viewer of a Raspberry Pi terminal. The terminal is split into two windows: 'publisher' and 'subscriber'. The 'publisher' window shows the execution of `python bmp280_mqtt_publish_json.py`, which outputs 'Message Published...' repeatedly. The 'subscriber' window shows the execution of `python mqtt_subscribe_json.py`, which outputs 'Subscribed to Topic: hello MQTT with QoS: (0,)' and then receives several JSON payloads containing sensor data like 'Pressure(hPa)' and 'Temp(C)'. A file manager window in the background shows the directory `/home/pi/Documents/paho` containing files like `P280.py`, `measure.py`, and `start.sh`.

```
# cd /home/pi/Documents/paho
# 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 bmp280_mqtt_publish_json.py 送信側立ち上げ
```



## AWS IoT開発

### ⑥. センサ情報リアルタイム表示

#### AWS DynamoDBでの確認

- Piでその実施した内容をDynamoDBで確認
- AWSのトップから>DynamoDBを選択
- テーブル>設定したテーブル>概要

The screenshot shows the AWS IAM console interface for a DynamoDB table named 'pi1\_db'. The '概要' (Summary) tab is active, and the table name 'pi1\_db' is highlighted in the table list. A red box highlights the table name and primary keys. A red arrow points to the 'ハッシュ、レンジの設定値' (Hash, Range Settings) section.

項目	値
テーブル名	pi1_db
プライマリパーティションキー	deviceid (文字列)
プライマリソートキー	timestamp (文字列)
有効期限 (TTL) 属性	無効 TTLの管理
テーブルの状態	有効
作成日	2018年1月31日 13:50:01 UTC+9
プロビジョニングされた読み込みキャパシティユニット	5 (Auto Scaling 無効)
プロビジョニングされた書き込みキャパシティユニット	5 (Auto Scaling 無効)
最後の減少時刻	-
最後の増加時刻	-
ストレージ容量 (バイト単位)	28.95 KB
項目数	211
リージョン	US East (Ohio)
Amazon リソースネーム (ARN)	arn:aws:dynamodb:us-east-1:123456789012:table/pi1_db

AWSのトップ画面からDynamoDBへ

ハッシュ、レンジの設定値





# AWS IoT開発

## ⑥. センサ情報リアルタイム表示

### IAMでのユーザ追加

- AWSTopp画面からIAMへ
- ユーザ>ユーザ追加

aws サービス リソースグループ spectrum-tech グローバル サポート

IAM の検索

ユーザーを追加 ユーザーの削除

ダッシュボード

グループ

ユーザー

ロール

ポリシー

ID プロバイダー

アカウント設定

認証情報レポート

暗号化キー

Q ユーザー名またはアクセスキーでユーザーを検索 2件の結果を表示中

ユーザー名	グループ	アクセスキーの古さ	パスワードの古さ	最後のアクティビティ	MFA
<input type="checkbox"/> dynamo_access	なし	4日間	4日間	今日	有効でない
<input type="checkbox"/> masa1	なし	33日間	なし	4日間	有効でない

**IoT, kyotoからのアクセス用にユーザを読み取り専用で作成します。**

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



## AWS IoT開発

### ⑥. センサ情報リアルタイム表示

#### リアルタイム表示

- IoT. Kyoto様の利用>グラフ
- Deviceid: BMP280 (pythonで書き込み済)
- 温度表示

<https://vis.iot.kyoto/>





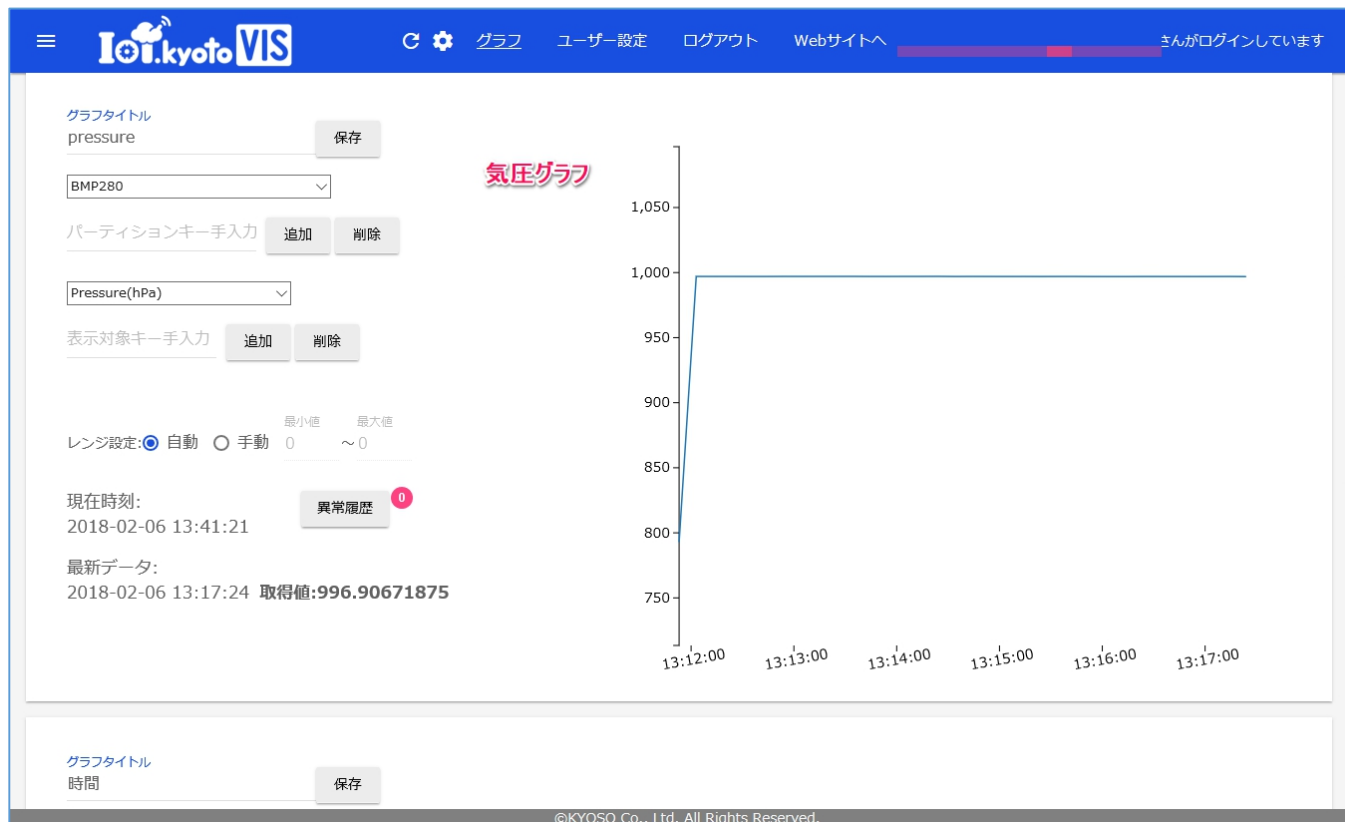
# AWS IoT開発

## ⑥. センサ情報リアルタイム表示

### リアルタイム表示

- IoT. Kyoto様の利用>グラフ
- Deviceid: BMP280 (pythonで書き込み済)
- 気圧表示

<https://vis.iot.kyoto/>





## AWS IoT開発

### ⑥. センサ情報リアルタイム表示

#### Cloudwatchの使い方

- トラブル時のログ確認ができます。
- AWSのトップ画面 > Cloudwatch

The screenshot displays the AWS CloudWatch dashboard interface. On the left, a navigation menu includes 'CloudWatch ダッシュボード' (highlighted with a red box), 'アラーム', 'イベント', 'ログ', and 'メトリクス'. The main content area is divided into three sections: 'メトリクスの要約' (Metrics Summary) with a search bar, 'アラームの要約' (Alarm Summary) featuring a warning message about alarm configurations and two line graphs for database capacity limits, and 'サービス状態' (Service Status) showing 'Amazon CloudWatch Service' as 'operating normally'. A red text annotation '使用しているクラウドの状態が確認できます。' is placed over the graphs.