

Azure IoT テクニカルワークショップ

Global Blackbelt Sales Microsoft Corporation

Oct-2018

Table of Contents

このハンズオンについて	1
概要	1
必要な準備	1
ハンズオンで構築するIoTサービスのについて	1
演習: Cloud Shell (Bash)の設定	2
タスク1: Cloud Shellの初期設定	2
演習: Azure環境の準備	3
タスク1 : Azureポータルへのアクセス	3
タスク2 : リソースグループの作成	3
演習: IoT Hubとデバイスの接続	6
タスク1 : IoT Hubの作成	6
タスク2 : IoT Hubへのデバイス登録	10
タスク3 : IoT Hubデバイスクライアントの開発	13
タスク4 : IoT Hubでのデータ受信確認	14
演習: Time Series Insightsによるデータの可視化	16
タスク1 : Time Series Insightsの作成	16
タスク2 : Time Series Insightsの設定	18
タスク3 : Time Series Insightsでのデータ表示とカスタマイズ	22
演習: Stream Analyticsによるリアルタイムデータ加工	25
タスク1 : Stream Analyticsの作成	25
タスク2 : Stream Analyticsの入力設定	27
タスク3 : Stream Analyticsの出力設定 (ストレージへの出力)	29
タスク4 : Stream Analyticsのクエリ設定 (ストレージへの出力)	34
タスク5 : Stream Analyticsの出力設定 (SQLデータベースへの出力)	37
タスク6 : Stream Analyticsのクエリ設定 (SQLデータベースへの出力)	42
演習 : Power BIでのデータ表示	45
タスク1 : Stream Analyticsの出力設定 (PowerBIへの出力)	45
演習: Azure IoT Edge デバイスの作成	55
タスク1 : Linux VMのデプロイ	55
タスク2 : コンテナランタイムのインストール	59
タスク3 : IoT Edgeデバイスの追加	61
タスク4 : IoT Edgeランタイムのインストール	63
タスク5 : IoT Edgeデバイスにシミュレーションモジュールを追加	65

このハンズオンについて

概要

このハンズオンでは、Microsoft

Azure IoTサービスを使用して
IoTアプリケーション（データの収集／加工／視覚化）をチュートリアル形式で構築します。Azure
IoTアーキテクチャをハンズオン形式で理解し、実現するシステムの構成をイメージできるようになるこ
とを目的とします。

必要な準備

Webブラウザから無線LANで接続可能なPCを必ず準備してください。

下記リンクから無償環境を取得いただけます。

<https://azure.microsoft.com/ja-jp/free/>

Power BIの環境を事前に取得の上、ご参加いただきますようお願いします。

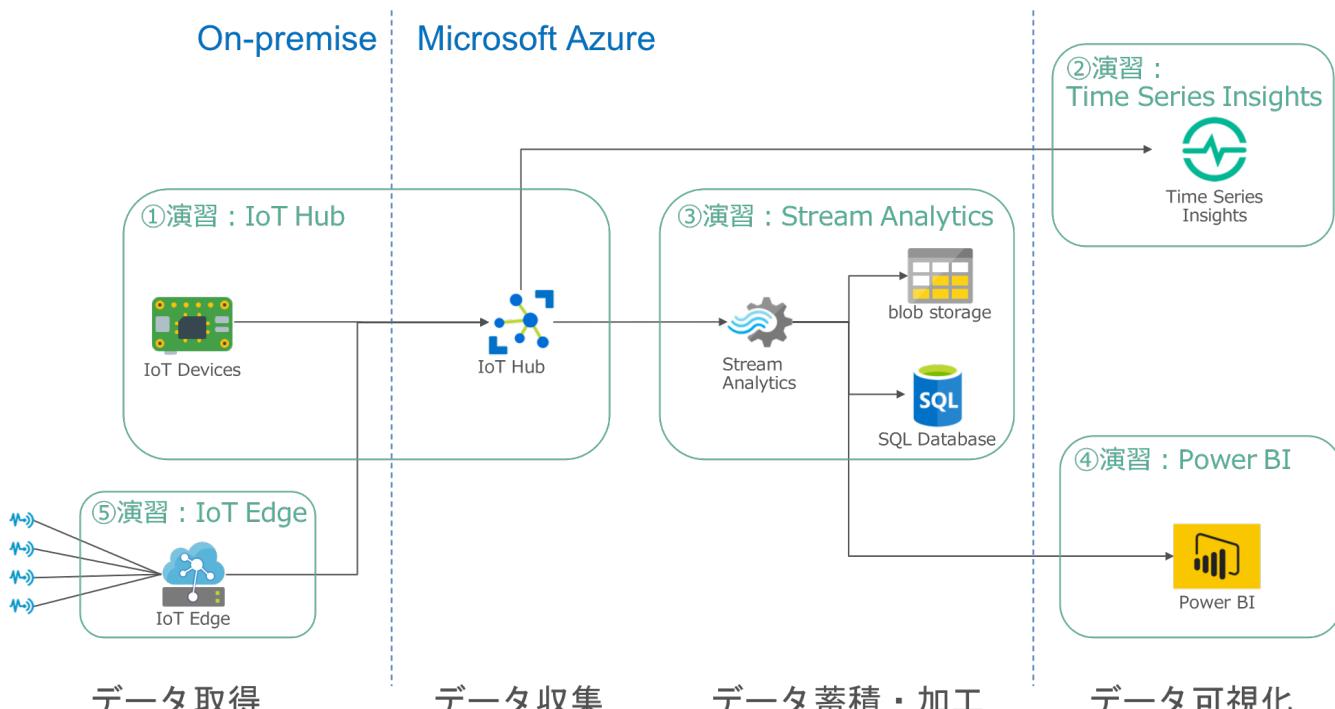
下記リンクから評価版の環境を取得いただけます。

<https://powerbi.microsoft.com/ja-jp/documentation/powerbi-service-self-service-signup-for-power-bi/>

下記リンクに本ハンズオンで使用するコードやコマンドを掲載していますのでご活用ください。

<https://msjpiot.blob.core.windows.net/docs/Snippet.txt>

ハンズオンで構築するIoTサービスのついて



演習: Cloud Shell (Bash)の設定

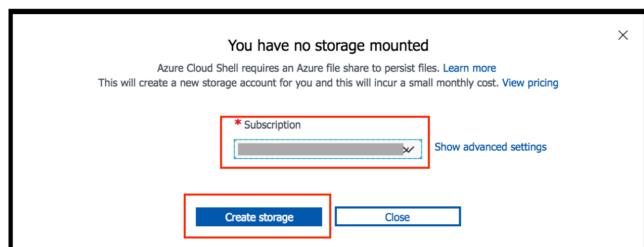
Cloud Shell (Bash) 環境を設定します。

タスク1: Cloud Shellの初期設定

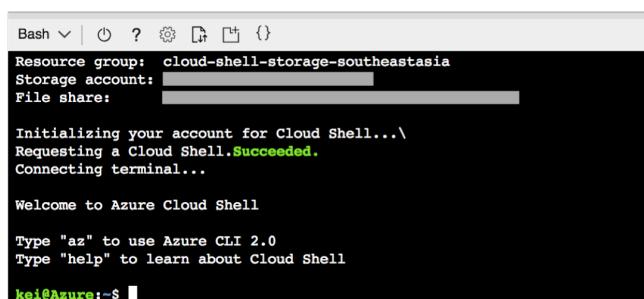
- [>] ボタンをクリックしてCloud Shellを開始します。
- Azureポータル画面下部のダイアログで [Bash(Linux)] を選択します。



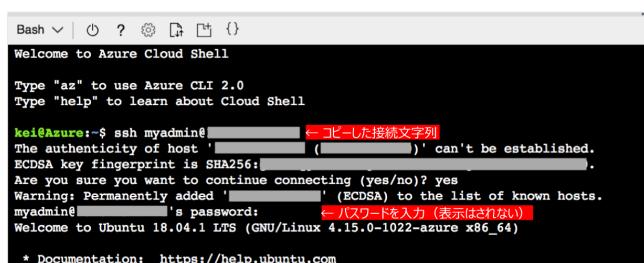
- Cloud Shell用のストレージアカウントに利用するサブスクリプションを選択し[Create storage]ボタンをクリックします。



- ストレージの作成が完了し、Bashが起動されることを確認します。



- VMにSSHでログインする場合は、接続文字列を入力してログインします。



演習:Azure環境の準備

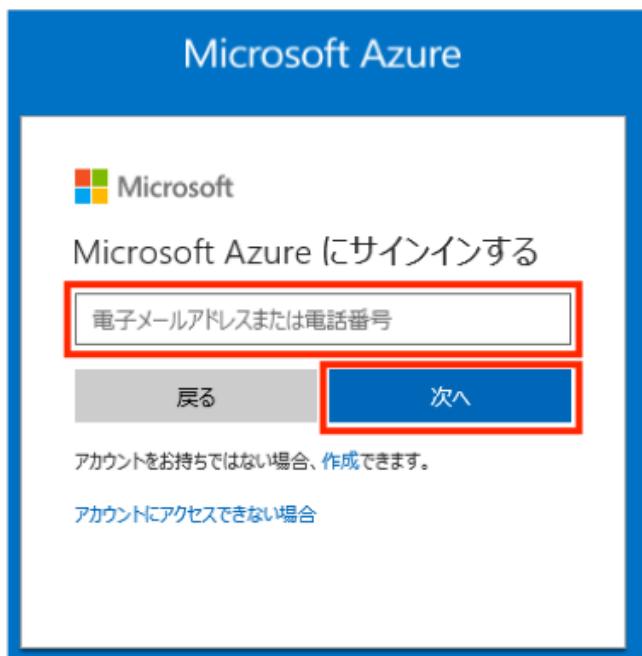
タスク1: Azureポータルへのアクセス

このタスクでは、Azureを操作するために、WebブラウザでAzureポータルにアクセスします。

1. ブラウザでAzure管理ポータルにアクセスします。

<https://portal.azure.com>

2. ログイン画面に取得いただいたマイクロソフトアカウントの[メールアドレス]と[パスワード]を入力して、[サインイン]ボタンをクリックします。



3. 管理ポータルにログインできたことを確認してください。

タスク2: リソースグループの作成

このタスクでは、リソースグループを作成します。リソースグループでは関連するサービスを同一のリソースグループ内にまとめて管理することができます。

Microsoft Azure プレビュー バグの報告 リソース、サービス、ドキュメントの検索

ダッシュボード リソースの作成 リソース グループ 新しいダッシュボード アップロード ダウンロード

リソース グループ すべてのサブスクリプション 更新

[アイコン]	東日本
[アイコン]	米国西部
[アイコン]	東日本
[アイコン]	東日本
[アイコン]	東日本
[アイコン]	米国東部
[アイコン]	西日本
[アイコン]	米国東部

表示数を増やす...

リソースの作成 すべてのサービス お気に入り ダッシュボード すべてのリソース リソース グループ App Service Function App SQL データベース Azure Cosmos DB Virtual Machines ロード バランサー

1. サイドメニューの[リソースグループ]をクリックします。

Microsoft Azure プレビュー リソースの作成 すべてのサービス お気に入り ダッシュボード すべてのリソース リソース グループ **App Service Function App SQL データベース Azure Cosmos DB Virtual Machines ロード バランサー**

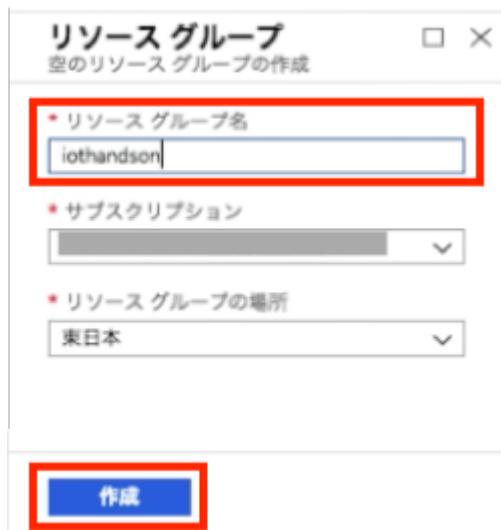
2. [+追加]をクリックします。リソースグループ作成ウィンドウが表示されます。

リソース グループ Microsoft **+ 追加** 列の編集 その他

3. 下記の項目を設定します。各設定項目を入力後、ウィンドウ下部にある[作成]をクリックします。

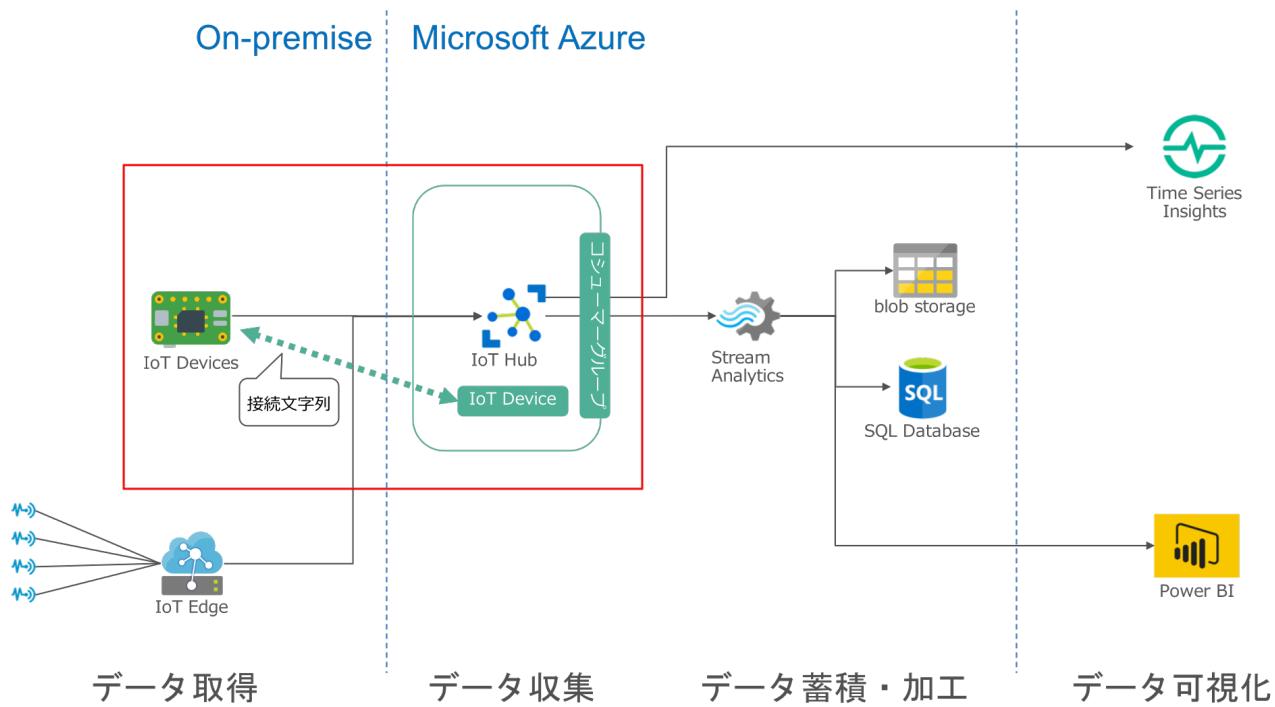
Table 1. 設定項目と設定値

設定項目	設定値
リソースグループ名	iothandson 任意の値でも構いません ただし、サブスクリプション内で一意の名前にする必要があります 縁のチェックマークが表示されれば重複していません
サブスクリプション	ご用意いただいたAzureサブスクリプション（デフォルトのまま）
リソースグループの場所	東日本



- リソースグループ画面の[更新]をクリックします。設定したリソースグループが追加されていることを確認してください。





演習: IoT Hubとデバイスの接続

この演習では、Azure上にIoT Hubを作成し、デバイスからのメッセージを受信します。デバイスとしてRaspberry Pi オンライン シミュレーター(Webアプリケーション)を使用し、作成したIoT Hubに温度データと湿度データを送信します。

NOTE

実際のPCやデバイスからデータを送信する場合、下記リンクに各開発言語/プラットフォーム毎 (Node.js, .NET, Java, Python, iOS) の手順が掲載されています。

<https://docs.microsoft.com/ja-jp/azure/iot-hub/>

タスク1：IoT Hubの作成

このタスクでは、デバイスからのデータを受信するIoT Hubを作成します。

1. サイドメニューの[リソースグループ]をクリックし、リソースグループ一覧を表示します。
2. 「演習：Azure環境の準備」で作成したリソースグループの名前をクリックします。指定したリソースグループのリソース一覧画面が表示されます。（現在は空です）
3. [+追加]をクリックします。Everything ウィンドウが表示されます。



4. “IoT Hub”と入力し、Enter キーを入力します。検索結果が表示されます。[IoT Hub]をクリックします。

名前	公開元	カテゴリ
IoT Hub	Microsoft	モノのインターネット ...
IoT Hub デバイス プロビジョニング サービス	Microsoft	モノのインターネット ...

5. IoT Hub の詳細説明ウィンドウが表示されます。ウィンドウ下部の[作成]をクリックします。

6. IoT Hubの作成ウィンドウが表示されます。各設定項目を入力後、ウィンドウ下部にある[Next: Size and scale]をクリックします。

Table 2. 設定項目と設定値

設定項目	設定値
サブスクリプション	ご用意いただいたAzureサブスクリプション（デフォルトのまま）
リソースグループ	既存のものを使用 「演習： Azure環境の準備」で作成したリソースグループ。 iothandson
場所	東日本（デフォルトのまま）
名前	任意の名前 例) iothub-xx IoT Hubの名前はサービス URLとして使用されるため、 Microsoft Azure全体で一意の名前を指定する必要があります

IoT hub
Microsoft

Basics Size and scale Review + create

Create an IoT Hub to help you connect, monitor, and manage billions of your IoT assets. [Learn More](#)

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription:

* Resource Group: 新規作成

* Region: 東日本

* IoT Hub Name: iothub-01809

Review + create Next: Size and scale > Automation options

7. Size and scale ウィンドウで設定を確認して、ウィンドウ下部の[Review + create]をクリックします。

Device-to-cloud/パーティションは、Advancedの項目の表示で確認できます。

Table 3. 設定項目と設定値

設定項目	設定値
価格とスケールティア	S1 – Standard (デフォルトのまま, IoT Edge はStandardで利用可能)
IoT Hub ユニット	1 (デフォルトのまま)
Device-to-cloud /パーティション	4 (デフォルトのまま)

8. [Review + create] ウィンドウで[Create]ボタンをクリックしてIoT Hubを作成します。

9. デプロイが開始されると、デプロイの進行状況が表示されます。

10. デプロイが完了すると、デプロイの進捗状況に「デプロイが完了しました」と表示されます。[リソースに移動]をクリックし、設定したIoT Hubが作成されていることを確認します。

11. IoT Hubのメニューから[Built-in endpoints]をクリックします

IoT Hub Overview

リソース グループ (変更) iothandson Hostname iothub-ko.azure-devices.net

状態 Active Pricing and scale tier S1 - Standard

場所 東日本 Number of IoT Hub units 1

サブスクリプション (変更) Microsoft Azure 社内従量課金プラン

サブスクリプション ID

Built-in endpoints

Need a way to provision millions of devices? IoT Hub Device Provisioning Service enables zero-touch, just-in-time provisioning to the right IoT hub without requiring human intervention.

12. [Built-in endpoints] ウィンドウの[CONSUMER GROUPS]に、既定の \$Default に加えて2つの消費者グループを追加します。(消費者グループの名前の例 : timeseriesinsights , streamanalytics)

消費者グループ(コンシューマーグループ)は、以降のステップでIoT Hubに送信されたデバイスデータを取得するサービス(Stream Analytics, Time Series Insights)が使用するものです。

Pricing and scale

Operations monitoring

IP Filter

Certificates

Built-in endpoints

Events

Event Hub-compatible name

Event Hub-compatible endpoint

Retain for

Consumer Groups

CONSUMER GROUPS

\$Default

timeseriesinsights

streamanalytics

タスク2：IoT Hubへのデバイス登録

このタスクでは、IoT Hubにデータを送信するデバイスを新規登録します。デバイスがIoT Hubにデータを送信するためには、事前にIoT Hubにデバイスを登録しておく必要があります。IoT Hub上でデバイスを新規登録し、デバイスIDとデバイスキー（認証キー）を発行します。

1. タスク1で作成したIoT Hubの設定ウィンドウを表示します。

リソースの作成

すべてのサービス

お気に入り

ダッシュボード

すべてのリソース

リソース グループ

App Service

Function App

SQL データベース

Azure Cosmos DB

Virtual Machines

ロード バランサー

ストレージ アカウント

リソース グループ

iothandson01

名前

iohub-201809

アクティビティ ログ

アクセス制御 (IAM)

タグ

イベント

設定

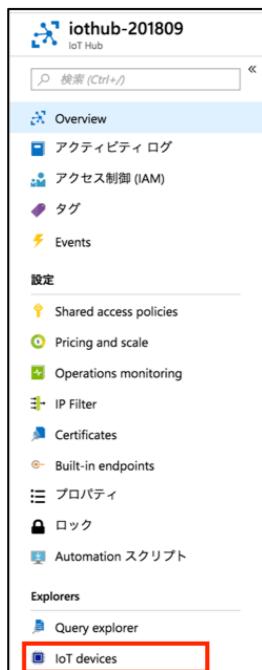
クイック スタート

リソース コスト

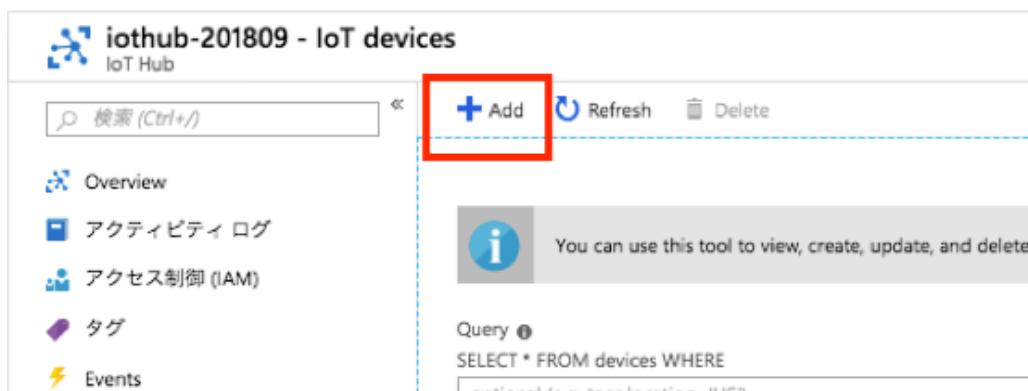
デプロイ

ポリシー

2. IoT HubのメニューのExplorersのセクションから[IoT devices]をクリックします。



3. ウィンドウ上部の[+Add]をクリックします。



4. [デバイスの作成]ウィンドウにsampledeviceと入力し、[Save]をクリックし、デバイスを登録します。ここで登録する文字列がデバイスIDとなります。

Create a device

Learn more about creating devices

* Device ID ⓘ
sampledevice

Authentication type ⓘ
Symmetric key X.509 Self-Signed X.509 CA Signed

* Primary key ⓘ
Enter your primary key

* Secondary key ⓘ
Enter your secondary key

Auto-generate keys ⓘ

Connect this device to an IoT hub ⓘ
Enable Disable

Parent device (Preview) ⓘ
No parent device
Set a parent device

Save

5. [デバイスエクスプローラー]のデバイス一覧に、登録したデバイスが追加されていることを確認します。追加したデバイスをクリックすると、[デバイスの詳細]ウィンドウが表示されます。

iotHub-201809 - IoT devices

Add Refresh Delete

You can use this tool to view, create, update, and delete devices on your IoT Hub.

Learn more about the device twin query syntax

SELECT * FROM devices + Add a WHERE clause Write query text

Run

DEVICE ID	STATUS	LAST ACTIVITY	LAST STATUS...	AUTHENTICA...	CLOUD TO DE...
sampledevice	Enabled	Sas	0		

6. [デバイスの詳細]ウィンドウで、デバイスキーや接続文字列の確認や、デバイスツインやデバイスへのメッセージ送信、ダイレクトメソッドが利用できます。ここでは”Connection string (primary key)”の値をコピーしておきます。次のタスクで、コピーした接続文字列をデバイスのプログラムコード内に設定します。

The screenshot shows the 'Device details' page for a device named 'SimpleDevice'. It includes fields for 'Device Id', 'Primary key', 'Secondary key', and 'Connection string (primary key)' (which is highlighted with a red box). Below these are sections for connecting to an IoT hub ('Enable' or 'Disable') and setting a parent device ('No parent device' or 'Set a parent device'). At the bottom, there are tabs for 'Module identities' and 'Configurations', with the former being active.

タスク3：IoT Hubデバイスクライアントの開発

このタスクでは、Raspberry Piオンラインシミュレーターを使ってIoT Hubにデータを送信します。本シミュレーターはRaspberry PiをシミュレートしたWebアプリケーションで、Node.jsでプログラムコードを記述することでIoT Hubにデータを送信することができます。

1. Raspberry Pi オンライン シミュレーターの設定

<https://docs.microsoft.com/ja-jp/azure/iot-hub/iot-hub-raspberry-pi-web-simulator-get-started>

The screenshot shows the 'Raspberry Pi オンライン シミュレーター の Azure IoT Hubへの接続 (Node.js)' guide. It includes a note about the date (2018/04/11) and author (共同作成者). A dropdown menu shows 'Raspberry Pi Web シミュレーター'. The main content explains the connection process between the simulator and the IoT hub, showing a diagram where data flows from the 'Raspberry Pi Simulator' to the 'Azure IoT hub'. A large blue button at the bottom is labeled 'START RASPBERRY PI SIMULATOR' (which is also highlighted with a red box).

- 上記URLをブラウザで開き、[START RAPBERRY PI SIMULATOR]をクリックします。
- [Step1 Raspberry Pi Web シミュレーターの概要]ウィンドウが開きます。[次のステップ]をクリックします。
- [Step2 IoT Hubの作成]ウィンドウが表示されます。[次のステップ]をクリックします。
- [Step3 Pi Web シミュレーターでのサンプルアプリケーションの実行]ウィンドウが表示されます。[私は理解して]をクリックします。
- [Raspberry Pi Azure IoT Online Simulator]ウィンドウが開きます。
- 15行目のconnectionStringの'[Your IoT hub device connection string]'を、作成したIoT Hubの接続文字列に変更します。記載方法は[]までを書き換えます。'コピーした接続文字列'とな

ります。

変更前

```
const connectionString = '[Your IoT hub device connection string]';
```

変更後

```
const connectionString = 'HostName=iothub0910-001.azure-devices.net;DeviceId=sampledevice;SharedAccessKey=xxxxxxxxxxxxxxxxxxxxxx';
```

- g. 23行目～37行目にある **function getMessage(cb)**
を下記に書き換えます。シミュレーションデバイスが生成したデータ(機械: 温度/圧力, 大気: 温度/湿度)をJSONで返します。

```
function getMessage(cb) {
    messageId++;
    sensor.readSensorData()
        .then(function (data) {
            cb(JSON.stringify({
                messageId: messageId,
                deviceId: 'Raspberry Pi Web Client',
                machine : { temperature : data.temperature_C, pressure : 10 },
                ambient : { temperature : data.temperature_C, humidity : data.humidity }
            }), data.temperature_C > 30);
        })
        .catch(function (err) {
            console.error('Failed to read out sensor data: ' + err);
        });
}
```

- h. 画面下部の[Run]ボタンをクリックします。画面左のLEDが点滅していれば動作しています。
またコンソールウィンドウに送信したテレメトリデータが表示されることを確認します。

エラーが発生せず、下記のように表示されていれば成功です。Webアプリケーションを閉じるか
[Stop]ボタンをクリックするまでテレメトリデータは送信され続けます。

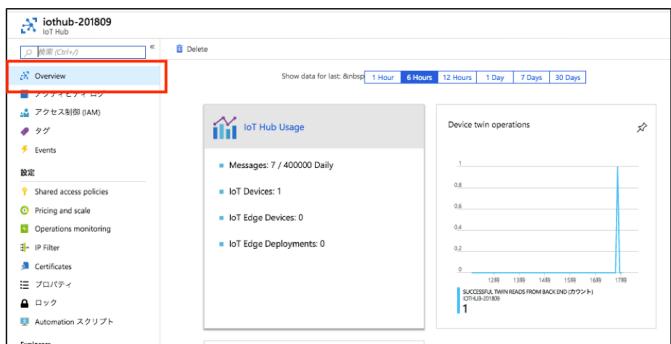
```
Sending message: {"messageId":1,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":23.61089154634202,"pressure":10},"ambient":{"temperature":23.61089154634202,"humidity":73.80948916321065}}
>
Message sent to Azure IoT Hub
>
```

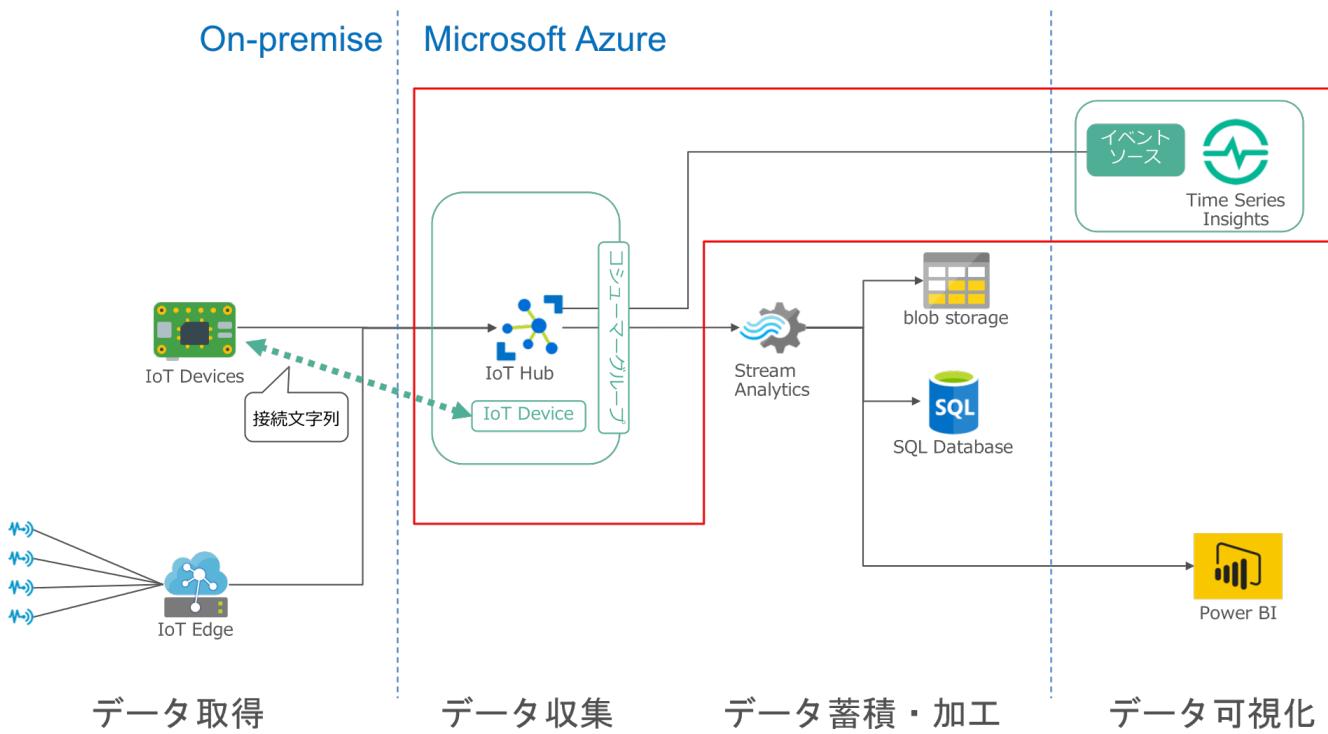
タスク4：IoT Hubでのデータ受信確認

このタスクでは、作成したIoT

Hubの[概要]

]をクリックし、使用状況を表示します。メッセージがカウントアップされていればデバイスからのテレメトリデータが受信できています。





演習: Time Series Insightsによるデータの可視化

この演習では、IoT Hubで受信したデータをTime Series Insightsで表示します。

Time Series Insightsの詳細は下記URLに記載があります。

NOTE

<https://docs.microsoft.com/ja-jp/azure/time-series-insights/time-series-insights-overview>

Time Series Insights は時系列データを念頭に設計されたデータベースです。ほぼリアルタイムでのデータ探索が可能です。また Time Series Insights REST API を使用して新しいカスタムソリューションを作成することもできます。既定のリテンション期間は31日間となっており、最大400日のデータリテンション期間を構成できます。

タスク1：Time Series Insightsの作成

このタスクでは、Azure上にTime Series Insightsを作成します。

1. 「演習: Azure環境の準備」で作成したリソースグループを表示し、ウィンドウ上部の[+追加]をクリックします。

2. Everythingウィンドウの検索ボックスで、"Time Series Insights"と入力しEnterキーを押します。
3. [Time Series Insights]をクリックします。

4. 表示された詳細ウィンドウ下部の[作成]をクリックします。

5. Time Series Insights environmentウィンドウの各設定値を入力後、ウィンドウ下部にある[作成]をクリックします。

Table 4. 設定項目と設定値

設定項目	設定値
環境名	任意の名前 例) tsi001
サブスクリプション	ご用意いただいたAzureサブスクリプション（デフォルトのまま）
リソースグループ	「演習：Azure環境の準備」で作成したリソースグループを選択
場所	米国西部（デフォルトのまま）
SKU	S1（デフォルトのまま）
Capacity	1（デフォルトのまま）



6. デプロイが開始されると、通知ウィンドウに表示されます。
7. デプロイが完了すると、通知ウィンドウに完了メッセージが表示されます。[リソースに移動]をクリックし、設定したTime Series Insightsが作成されていることを確認します。



タスク2：Time Series Insightsの設定

このタスクでは、作成したTime Series Insightsに、「演習：IoT Hubとデバイスの接続」で作成したIoT Hubをイベントソースに設定します。

1. タスク1で作成したTime Series Insightsの設定ウィンドウを表示します。
2. Time Series Insightsの設定メニューにある[イベントソース]をクリックします。ウィンドウ上部の [+追加]をクリックします。



3. [イベントソース]ウィンドウの各設定値を入力後、ウィンドウ下部にある[作成]をクリックします。

Table 5. 設定項目と設定値

設定項目	設定値
イベントソース名 Event source name	任意の名前 例) es001
ソース Source	IoT Hub
インポートオプション Import option	使用可能なサブスクリプションから... (デフォルトのまま) Use IoT Hub from available subscription
サブスクリプションID Subscription Id	ご用意いただいたサブスクリプション (デフォルトのまま)
IoT Hub名 IoT hub name	「演習:IoT Hubのデバイス接続」で作成したIoT Hub名 ex) iothub-xx
IoT Hubポリシー名 IoT hub policy name	自動入力のまま (iothubowner)
IoT Hubポリシーキー Iot hub policy key	自動入力のまま (***)
IoT Hub コンシューマグループ Iot hub consumer group	timeseriesinsights (IoT Hubで追加したコンシューマグループ)
Event serialization format	自動入力のまま (JSON)
Timestamp property name	空白のまま



4. [イベントソース]ウィンドウに追加したイベントソースが表示されていることを確認します。確認後、右上の[X]ボタンをクリックして[イベントソース]ウィンドウを閉じます。Time Series

Insights概要ウィンドウの”イベントソース”が ”1” となります。

The screenshot shows the Azure Time Series Insights environment overview page for 'timeseriesinsights001'. The left sidebar includes options like Overview, Logs, IAM, Tags, Settings (Lock, Automation Script), Configuration, Environment Topology, Data Access Policy, Event Sources, and Reference Data Sets. The main area displays basic information such as Resource Group (iothandson01), Location (East US), Creation Date (September 12, 2018), SKU (S1), Retention Policy (31 days), and a Data Access FQDN URL. A table titled 'Event Source' shows one entry: 'eventsource001' from an IoT Hub. The 'Event Sources' column is highlighted with a red box.

5. 以下のようなデータアクセスポリシーの警告が表示される場合、データアクセスポリシーを追加します。

The screenshot shows the Azure Time Series Insights environment overview page for 'timeseriesinsights001'. The left sidebar includes options like Overview, Logs, IAM, Tags, Settings (Lock, Automation Script), Configuration, Environment Topology, Data Access Policy, Event Sources, and Reference Data Sets. A yellow warning bar at the top states: '⚠ データアクセス ポリシーを追加してください。データアクセス ポリシーが定義されていない環境のデータにはアクセスできません。' (Please add a data access policy. You cannot access data in environments where a data access policy is not defined.)

Tim Series Insightsのメニューから[データアクセスポリシー]をクリックします。

The screenshot shows the Azure Time Series Insights environment menu for 'timeseriesinsights001'. The left sidebar includes options like Overview, Logs, IAM, Tags, Settings (Lock, Automation Script), Configuration, Environment Topology, and Data Access Policy. The 'Data Access Policy' option is highlighted with a red box.

- a. データアクセスポリシーウィンドウで、ユーザーが一件も登録されていないことを確認して[追加]をクリックします。

- b. [ユーザー役割の選択] ウィンドウで[ユーザーの選択]をクリックして、Azure管理ポータルにログインしたご自身のアカウント(Azure管理ポータルの右上に表示)を検索して追加します。

- c. [ユーザー役割の選択] ウィンドウで[ロールを選択する]をクリックして、[共同作成者]を選択して[OK]をクリックします。

- d. ユーザーとロールの選択後、[OK]をクリックします。

6. データ保持期間やキャパシティを設定する場合には、[構成]メニューをクリックします。

容量やデータリテンション期間（日）を必要に応じて設定します。

タスク3：Time Series Insightsでのデータ表示とカスタマイズ

このタスクでは、作成したTime Series Insightsエクスプローラーへアクセスし、IoT Hubで受信した温度データと湿度データを表示します。

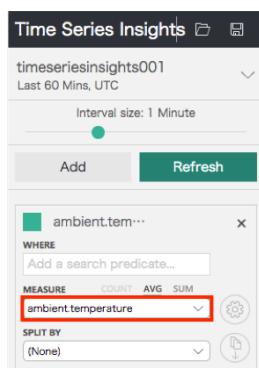
1. 作成したTime Series Insightsを表示します。

2. [概要]ウィンドウ上部の[環境を開く]をクリックします。[Time Series InsightsエクスプローラーのURL]がURLになります



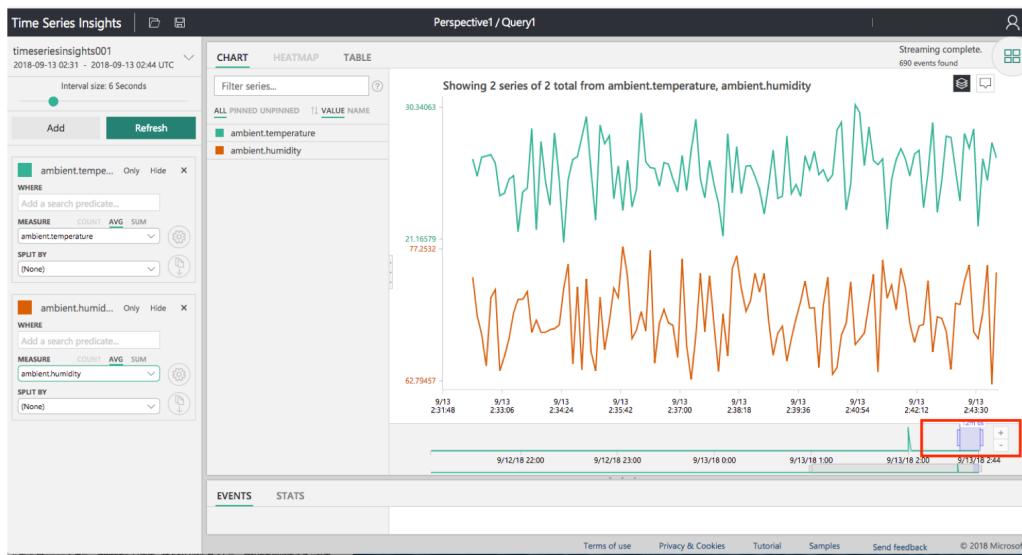
3. Time Series Insightsのポータル画面が表示されます。

4. 温度データを表示するために、サイドメニューのMEASUREプルダウンを”Events”から”ambient.temperature”に変更します。



5. 湿度データを追加するために、サイドメニューの[Add]をクリックし、MEASUREプルダウンを”Events”から”ambient.humidity”に変更します。

6. より詳細なデータを確認するために、表示期間を調整します。紫色の両端をドラッグアンドドロップし、範囲を調整します。ポータル左上のメニューからも範囲設定ができます。

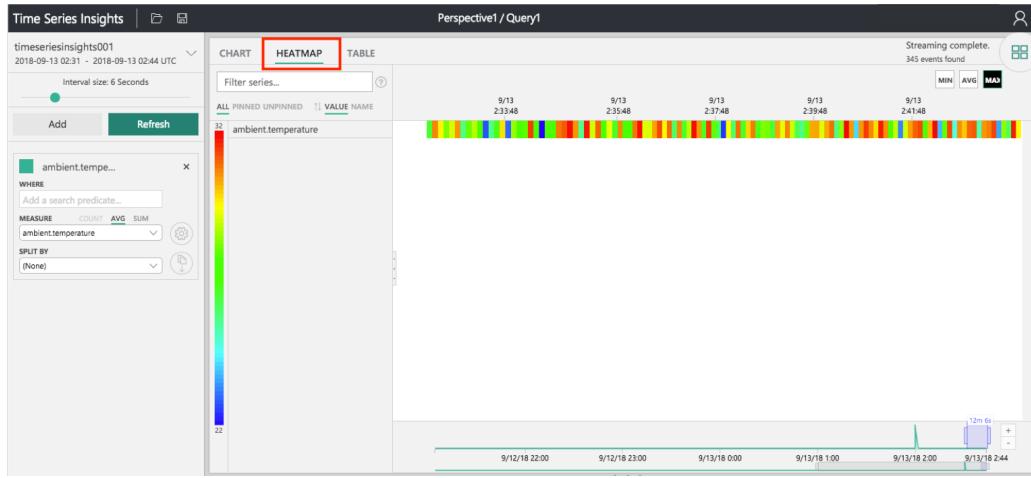


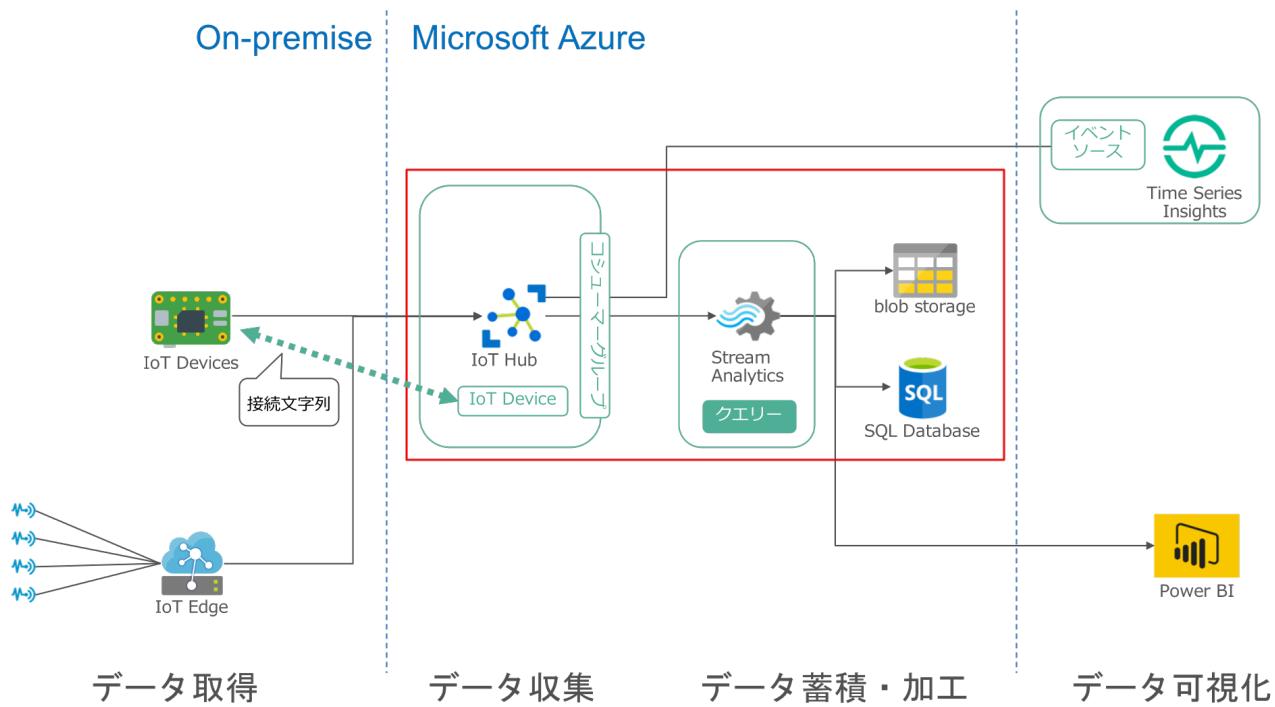
7. 左サイドメニューの[Interval]

size]

を設定すると、グラフの描画粒度を調整することができます。Interval sizeごと平均の値がグラフに描画されています。

8. 表示するプロパティが1つの場合、ヒートマップを表示することができます。”ambient.humidity”のグラフ表示設定を削除し、[HEATMAP]タブをクリックします。ヒートマップは多くのデバイスを表示する際のデータ傾向把握に有効です。





演習: Stream Analyticsによるリアルタイムデータ加工

この演習では、Stream Analyticsにより、リアルタイム分析とBlob Storageへのデータ格納を行います。様々なリアルタイムデータ処理を実現することができます。ここでは、個別データと1分単位のサマリデータを出力するように設定します。

NOTE

Stream Analyticsの詳細は、下記URLに記載があります。

<https://docs.microsoft.com/ja-jp/azure/stream-analytics/>

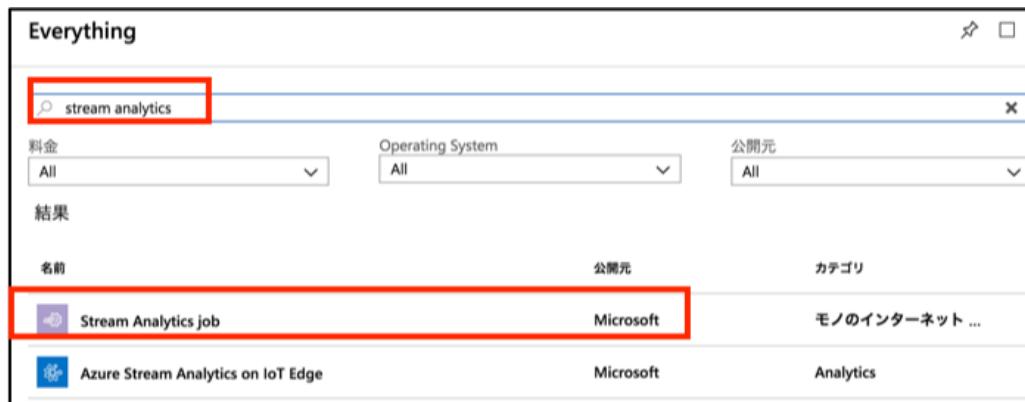
タスク1：Stream Analyticsの作成

このタスクでは、Azure上にStream Analyticsを作成します。

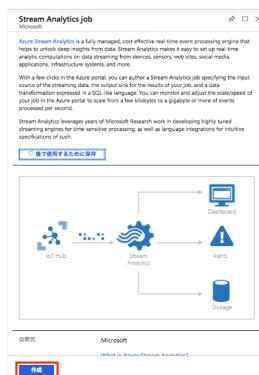
- 「演習：Azure環境の準備」で作成したリソースグループを表示し、ウィンドウ上部の[+追加]をクリックします。



- Everythingウィンドウの検索ボックスで、"Stream Analytics"と入力しEnterキーを押します。



3. [Stream Analytics job]をクリックし、表示された詳細ウィンドウ下部の[作成]をクリックします。



4. [新しい Stream Analytics ジョブ] ウィンドウの各設定値を入力後、ウィンドウ下部にある[作成]をクリックします。

Table 6. 設定項目と設定値

設定項目	設定値
ジョブ名 Job name	任意の名前 例) asajob001
サブスクリプション	ご用意いただいたAzureサブスクリプション（デフォルトのまま）
リソースグループ Resource group	既存のものを使用 「演習:Azure環境の準備」で作成したリソースグループ。 ex) iothandson
場所	東日本（デフォルトのまま）
ホスティング環境 Hosting environment	Cloud（デフォルトのまま）
ストリーミングユニット Streaming units	3（デフォルトのまま）



5. デプロイが完了すると、通知ウィンドウに完了メッセージが表示されます。[リソースに移動]をクリックし、設定したStream Analyticsが作成されていることを確認します。

タスク2：Stream Analyticsの入力設定

このタスクでは、作成したStream Analyticsに[入力](Stream Analyticsへの入力データ)を設定します。

- 「タスク：Stream Analyticsの作成」で作成したStream Analyticsの設定ウィンドウを表示します。

[サイドメニュー] > [リソースグループ] > [作成したリソースグループ] > “作成したStream Analyticsの名前”

- Stream Analyticsの設定メニューにある[Inputs]をクリックし、ウィンドウ上部の[+Add stream input]-[IoT Hub]をクリックします。

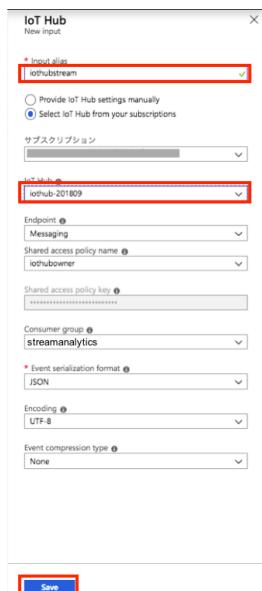
The screenshot shows the Azure Stream Analytics job interface for 'asajob001'. On the left, there's a sidebar with links like Overview, Activity Log, IAM, Tags, Troubleshoot, Settings, and Lock. Below that is the 'Job topology' section with 'Inputs' selected, indicated by a red box. In the main area, there are buttons for 'Add stream input' and 'Add reference input'. A dropdown menu is open under 'Add stream input', showing options: Event Hub, IoT Hub (which is highlighted with a red box), and Blob storage.

3. [新規入力] ウィンドウの各設定値を入力後、ウィンドウ下部にある[作成]をクリックします。

Table 7. 設定項目と設定値

設定項目	設定値
入力のエイリアス Input alias	任意の名前 指定した名前は後述のクエリの構文内で参照 例)IoTHubStream
インポートオプション ◦Provide IoT Hub settings manually •Select IoT Hub from your subscriptions	Select IoT Hub from your subscriptions (デフォルトのまま)
サブスクリプション	ご用意いただいたAzureサブスクリプション (デ フォルトのまま)
IoT Hub	「演習：IoT Hub とデバイスの接続」で作成したIoT Hubを選択
エンドポイント Endpoints	Messaging
共有アクセスポリシー名 Shared access policy name	iothubowner (デフォルトのまま)
共有アクセスポリシーキー Shared access policy key	デフォルトのまま
コンシューマーグループ Consumer group	「演習：IoT Hub とデバイス接続」で追加したStream Analytics用のコンシューマーグループ streamanalytics
イベントシリアル化形式 Event serialization format	JSON (デフォルトのまま)
エンコード+ Encoding	UTF-8 (デフォルトのまま)

設定項目	設定値
イベントの圧縮タイプ Event compression type	None



4. 通知ウィンドウに「入力のテスト中」と表示されます。「接続テストが成功しました」と表示されれば、入力の追加完了です。

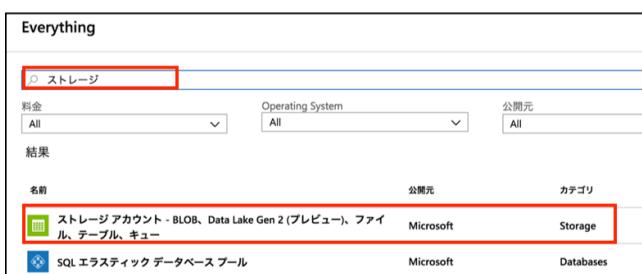
タスク3：Stream Analyticsの出力設定（ストレージへの出力）

このタスクでは、作成したStream Analyticsに出力を設定します。デバイスから送信された全データを Azure Blobストレージに保存します。

1. 「演習：Azure環境の準備」作成したリソースグループを表示し、ウィンドウ上部の[+追加]をクリックします。



2. Everythingウィンドウの検索ボックスで、”ストレージ”と入力しEnterキーを押します。[ストレージアカウント]をクリックします。



3. 表示された詳細ウィンドウ下部の[作成]をクリックします。



4. 下記のとおり、各設定値を入力後、ウィンドウ下部にある[作成]をクリックします。

Table 8. 設定項目と設定値

設定項目	設定値
基本タブ	
サブスクリプション	ご準備いただいたサブスクリプション
リソースグループ	既存のものを使用 「演習:Azure環境の準備」で作成したリソースグループ ex) iothandson
名前	任意の名前. Microsoft Azure全体で一意の名前 例) iotstorage
場所	東日本
パフォーマンス	Standard (デフォルトのまま)
アカウントの種類	StorageV2 (汎用 v2)
レプリケーション	ローカル冗長ストレージ (デフォルトのまま)
アクセス層	ホット (デフォルトのまま)
詳細タブ	
安全な転送が必須	無効
仮想ネットワーク	無効
Data Lake Storage Gen2 (プレビュー)	何もない

ストレージ アカウントの作成

基本 詳細 タグ 確認および作成

Azure Storage は、高可用性、セキュリティ、耐久性、スケーラビリティ、冗長性を備えたクラウドストレージを提供する Microsoft が管理するサービスです。Azure Storage には、Azure Blob (オブジェクト)、Azure Data Lake Storage Gen2、Azure Files、Azure Queues、Azure Tables が含まれます。ストレージアカウントのコストは、使用量と、下で選ぶオプションに応じて決まります。 [詳細情報](#)

プロジェクトの詳細

デプロイされているリソースとコストを管理するサブスクリプションを選択します。フォルダーのようなリソース グループを使用して、すべてのリソースを整理し、管理します。

* サブスクリプション

* リソース グループ iothandson iothandson

[新規作成](#)

インスタンスの詳細

既定の展開モデルはリソース マネージャーであり、これは最新の Azure 機能をサポートしています。代わりに、従来の展開モデルを使った展開も選択できます。 [クラシック展開モデルを選択します](#)

* ストレージアカウント名 iotstorage iotstorage

ストレージアカウント名 'iotstorage' は既に使用されています。

* 場所 東日本

パフォーマンス Standard Premium

アカウントの種類 StorageV2 (汎用 v2)

レプリケーション ローカル冗長ストレージ (LRS)

アクセス層 (既定) クール ホット

確認および作成 [前へ](#) [次: 詳細 >](#)

ストレージ アカウントの作成

基本 詳細 タグ 確認および作成

セキュリティ

安全な転送が必須 無効 有効

仮想ネットワーク

許可するアクセス元 すべてのネットワーク 選択したネットワーク

すべてのネットワークで、このストレージアカウントにアクセスできます。 [詳細情報](#)

DATA LAKE STORAGE GEN2 (プレビュー)

階層構造の名前空間 無効 有効

確認および作成 [前へ](#) [次: タグ >](#)

ストレージ アカウントの作成

✓ 検証に成功しました

基本 詳細 タグ 確認および作成

基本

サブスクリプション	Microsoft Azure 社内従量課金プラン
リソース グループ	iothandson
場所	東日本
ストレージ アカウント名	iotstorage201809
デプロイ モデル	Resource Manager
アカウントの種類	StorageV2 (汎用 v2)
レプリケーション	ローカル冗長ストレージ (LRS)
パフォーマンス	Standard
アクセス層 (既定)	ホット

詳細

安全な転送が必須	有効
許可するアクセス元:	すべてのネットワーク
階層構造の名前空間	無効

作成 前へ 次へ Automation のテンプレート

5. リソースの作成が完了したら、[リソースに移動]をクリックします。

Microsoft.StorageAccount-20181022021204 - 概要

デプロイ 検索 (Ctrl+ /) 削除 キャンセル 再デプロイ 最新の情報に更新

概要 入力 出力 テンプレート

● デプロイが完了しました

リソースに移動

デプロイ名: Microsoft.StorageAccount-20181022021204
サブスクリプション: Microsoft Azure 社内従量課金プラン
リソース グループ: iothandson

展開の詳細 (ダウンロード)
開始時刻: 2018/10/22 2:33:19
時間: 31 秒
相関 ID: e629072e-1acc-493b-b3cc-a0799dbac76a

リソース 構造 状態 操作の詳細

iotstorage2 Microsoft... OK 操作の詳細

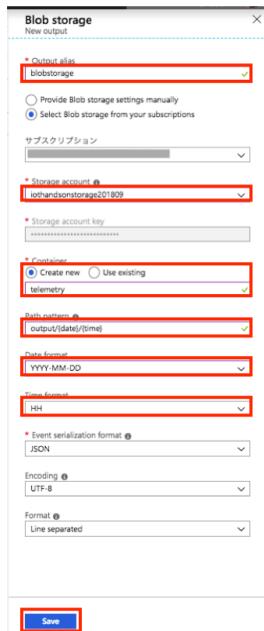
6. Stream Analyticsの設定メニューにある[出力]をクリックし、ウィンドウ上部の[+Add]-[Blob storage]をクリックします。

7. [新規出力]ウィンドウの各設定値を入力後、ウィンドウ下部にある[作成]をクリックします。

Table 9. 設定項目と設定値

設定項目	設定値
出力エイリアス Output alias	任意の名前。 指定した名前は後述のクエリの構文内で参照 例) blobstorage
インポートオプション ○Provide IoT Hub settings manually ●Select IoT Hub from your subscriptions	Select IoT Hub from your subscriptions (デフォルトのまま)
インポートオプション サブスクリプション	サブスクリプションからBlob Storageを選択する
ストレージアカウント Storage account	ご準備いただいたサブスクリプション
ストレージアカウントキー Storage accountkey	当タスクで作成したストレージアカウントを選択する
コンテナ指定方式 ●Create new ○Use existing	自動入力
コンテナー名 Container	Create new (デフォルトのまま)
パスパターン	任意の名前 例) telemetry
日付の形式 Date format	出力パスを設定する 例) output/{date}/{time}
時刻の形式 Time format	形式を選択する。ここでは YYYY-MM-DD を選択
イベントシリアル化形式 Event serialization format	形式を選択する
	JSON

設定項目	設定値
エンコード Encoding	UTF-8
フォーマット Format	Line separated



タスク4：Stream Analyticsのクエリ設定（ストレージへの出力）

このタスクでは、Stream Analyticsにストレージ出力するクエリを設定します。

1. Stream Analyticsの設定メニューにある[クエリ]をクリックします。

```

1 SELECT
2 *
3 INTO
4 [YourOutputAlias]
5 FROM
6 [YourInputAlias]
    
```

2. 右側のウィンドウ内のクエリを修正し、[Save]をクリックします。

```
SELECT * INTO [blobstorage] FROM [IoTHubStream]
```

The screenshot shows the 'asajob001 - Query' page in the Stream Analytics job configuration. The main area displays the query: `1 SELECT * INTO [blobstorage] FROM [IoTHubStream]`. A red box highlights the 'Save' button at the top right. On the left, there's a sidebar with navigation links: Overview, アクティビティ ログ, アクセス制御 (IAM), タグ, and 問題の診断と解決.

3. Stream Analyticsの設定メニューにある[Overview]をクリックし、ウィンドウ上部にある[Start]をクリックします。

The screenshot shows the 'asajob001' Stream Analytics job overview page. The 'Overview' link in the sidebar is highlighted with a red box. At the top, there are buttons for Start, Stop, and Delete. The 'Start' button is also highlighted with a red box. Below the buttons, it shows the resource group as 'iothandson01' and the status as 'Created'.

4. [ジョブの開始]ウィンドウが表示されたら、ウィンドウ下部の[Start]をクリックします。

The screenshot shows the 'Start job' dialog box for the 'asajob001' job. It has a 'Job output start time' section with 'Now' selected. At the bottom, there is a large blue 'Start' button, which is highlighted with a red box.

5. 通知ウィンドウで正常にStream Analyticsジョブが開始されたかを確認します。（1～2分かかります）

The screenshot shows the 'asajob001' Stream Analytics job overview page again. The 'Running' status is highlighted with a red box. To the right, a '通知' (Notification) window is open, showing a message: 'Streaming job started successfully.' and 'Started Streaming job 'asajob001' successfully.' by me a few seconds ago. A red box highlights this message.

6. 指定したストレージにテレメトリデータが保存されていることを確認します。

「演習：Azure環境の準備」で作成したリソースグループを表示し、作成したストレージアカウントを表示します。[ストレージアカウント]ウィンドウ内のサービス欄の[BLOB]をクリックします。

iothandsonstorage201809 リソース グループ (変更) パフォーマンス Standard
ストレージ アカウント iothandson01 状態 ブライマリ: 利用可能 レプリカ ローカル
場所 東日本 アカウント BLOB ス
サブスクリプション (変更)
サブスクリプション ID
タグ (変更)
タグを追加するにはここをクリック
サービス
BLOB 非構造化データの REST ベースのオブジェクトストレージ 詳細情報

7. 作成したコンテナ内に指定したパスが作成され、データが保存されていることを確認します。

iothandsonstorage201809 - BLOB リソース グループ (変更) + コンテナー 更新 削除
ストレージ アカウント: iothandsonstorage201809
名前
telemetry

8. ファイル名を選択後、ウィンドウ上部の[BLOBの編集]をクリックし、データを確認します。

output/2018-09-14/02/0_20e87363eb154ab5ba70d24f5200a0e0_1.json 保存 破棄 更新 ダウンロード 削除
概要 スナップショット BLOB の編集 SAS の生成
1 {"messageId":1,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":25.48613014821997,"pressure":10}, "ambient":
2 {"messageId":2,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":29.889242991815703,"pressure":10}, "ambient":
3 {"messageId":3,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":25.857484107037969,"pressure":10}, "ambient":
4 {"messageId":4,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":26.374606926239455,"pressure":10}, "ambient":
5 {"messageId":5,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":29.151924156140186,"pressure":10}, "ambient":
6 {"messageId":6,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":31.844064205655819,"pressure":10}, "ambient":
7 {"messageId":7,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":22.082956060440431,"pressure":10}, "ambient":
8 {"messageId":8,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":30.056753025494018,"pressure":10}, "ambient":
9 {"messageId":9,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":23.8367606397914,"pressure":10}, "ambient":
10 {"messageId":10,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":20.197792542602578,"pressure":10}, "ambient":
11 {"messageId":11,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":25.146795718991108,"pressure":10}, "ambient":
12 {"messageId":12,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":26.519494518057222,"pressure":10}, "ambient":
13 {"messageId":13,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":30.989853340777035,"pressure":10}, "ambient":
14 {"messageId":14,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":20.708662385589051,"pressure":10}, "ambient":
15 {"messageId":15,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":21.37819716341783,"pressure":10}, "ambient":
16 {"messageId":16,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":23.831764090442974,"pressure":10}, "ambient":
17 {"messageId":17,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":25.776524958498662,"pressure":10}, "ambient":
18 {"messageId":18,"deviceId":"Raspberry Pi Web Client","machine":{"temperature":26.256253542620133,"pressure":10}, "ambient":
...
プレビュー JSON

9. データを確認後、Stream Analyticsジョブを停止しておきます。

asajob001 Stream Analytics job 検索 (Ctrl+ /) Start Stop Delete
Overview Running

タスク5：Stream Analyticsの出力設定（SQLデータベースへの出力）

このタスクでは、作成したStream 1分間の平均値を保存します。

Analyticsに出力を追加します。SQLデータベースに

- 「演習:Azure環境の準備」で作成したリソースグループを表示し、ウィンドウ上部の[+追加]をクリックします。



- Everythingウィンドウの検索ボックスで、”SQL”と入力しEnterキーを押します。[SQL Database]をクリックし、表示された詳細ウィンドウ下部の[作成]をクリックします。

The screenshot shows the Azure portal's search interface for 'SQL'. The search bar has 'SQL' typed into it. Below the search bar, there are filters for '料金' (All), 'Operating System' (All), and '公開元' (All). The results list shows three items: 'Web App + SQL', 'SQL Server 2016 SP1 Enterprise on Windows Server 2016', and 'SQL Database'. The 'SQL Database' item is selected and highlighted with a red box. Below this, a detailed configuration dialog box is open for creating a new 'SQL Database'. The fields filled in are: Database name (sqldb2018001), Subscription (selected), Resource group (iothandson), Select source (Blank database), Server (sqlsvr2018001 (東日本)), and Collation (SQL_Latin1_General_CI_AS). At the bottom of this dialog, the 'Create' button is also highlighted with a red box.

- 下記のとおり、各設定値を入力後、ウィンドウ下部にある[作成]をクリックします。

Table 10. 設定項目と設定値

設定項目	設定値
データベース名	任意の名前 例) sqldb2018001
サブスクリプション	ご準備いただいたサブスクリプション（デフォルトのまま）
リソースグループ	既存のものを使用
ソースの選択	Blank database（デフォルトのまま）
サーバー	[必要な設定の構成]をクリックし、次の「新しいサーバーの作成」に記載の内容を入力。
新しいサーバーの作成	
サーバー名	任意の名前。Azure全体で一意の名前。 例) sqlsv2018001
サーバー管理者ログイン	任意の管理者IDを設定 例) dbadmin
パスワード	任意の管理者パスワードを設定 例) @dmin123
パスワードの確認	パスワードと同じ文字列を設定
場所	東日本（デフォルトのまま）
Azureサービスにサーバーへのアクセスを許可する	チェック（デフォルトのまま）
Advanced Thread Protection	Not now（デフォルトのまま）
～新しいサーバーの作成の項目はここまで～	
SQLエラスティックプールを使用しますか？	後で（デフォルトのまま）
価格レベル	[必要な設定の構成]をクリックし、[Basic]をクリック後、下部の[適用]をクリック
照合順序	SQL_Latin1_General_CI_AS（デフォルトのまま）

SQL Database

* Database name: sqldb2018001

* サブスクリプション: [選択]

* Resource group: iothandson

Select source: Blank database

Server: 必要な設定の構成

Want to use SQL elastic pool? Not now

Pricing tier: 必要な設定の構成

Collation: SQL_Latin1_General_CI_AS

Server

Create a new server

No servers found

New server

* Server name: sqlsv2018001

database.windows.net

* Server admin login: dbadmin

* Password: [redacted]

* Confirm password: [redacted]

* Location: 東日本

Allow Azure services to access server

Advanced Threat Protection

Start FREE Trial Not now

FREE trial period of 60 days, and then 1680 JPY/server/month.

Learn more

作成 Automation オプション

選択

SQL Database

* Database name: sqldb2018001

* サブスクリプション: [選択]

* Resource group: iothandson

Select source: Blank database

Server: sqlsv2018001 (東日本)

Want to use SQL elastic pool? Not now

Pricing tier: Standard S0: 10 DTUs, 250 GB

Collation: SQL_Latin1_General_CI_AS

作成 Automation オプション

4. リソースグループ内に作成したSQLデータベースが表示されていることを確認します。

5. SQLデータベースのメニューから[クエリエディタ]を選択し、
[OK]をクリックします。

[パスワード]を入力後、

Table 11. 設定項目と設定値

設定項目	設定値
承認の種類	SQL Server認証（デフォルトのまま）
ログイン	データベース作成時に指定したユーザー名 例) dbadmin（デフォルトのまま）
パスワード	データベース作成時に指定したパスワード 例) @dmin123

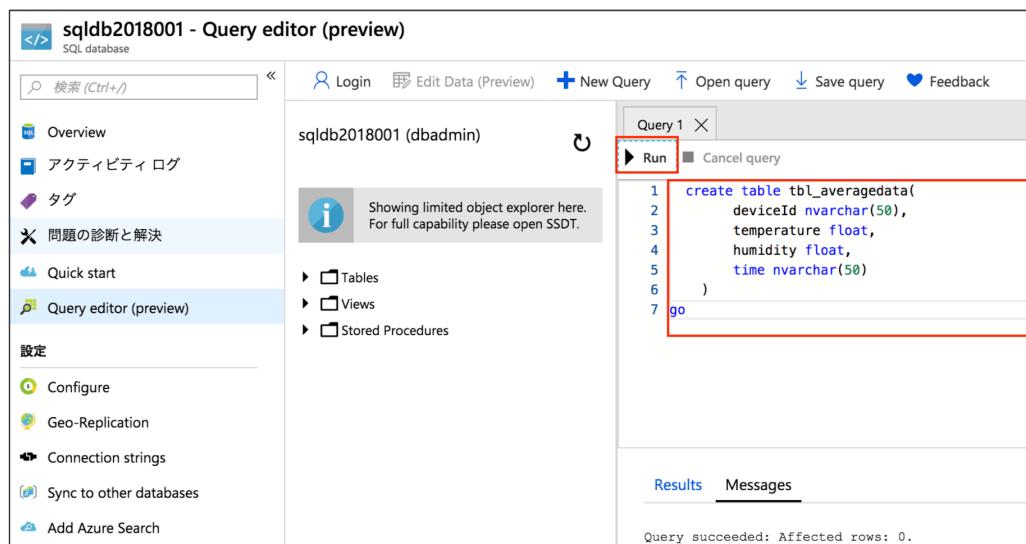
6. 1分平均値を格納するテーブル”tbl_averagedata”を作成します。

[クエリ1]に下記のSQLを入力し、[実行]をクリックします。テーブル一覧を更新し、テーブルが正常に作成されていることを確認します。

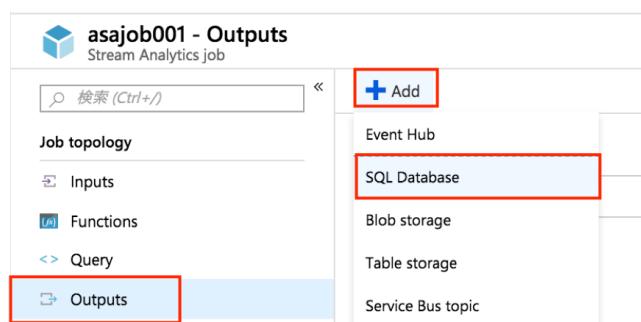
```

create table tbl_averagedata(
    deviceId nvarchar(50),
    temperature float,
    humidity float,
    time nvarchar(50)
)
go

```



- Stream Analyticsの設定メニューにある[Output]をクリックし、ウィンドウ上部の[+Add]-[SQL Database]をクリックします。

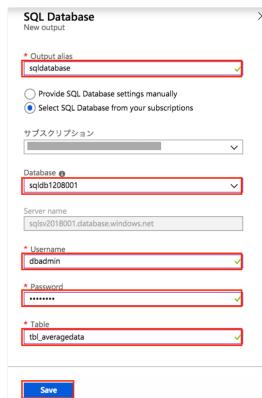


- [新規出力]ウィンドウの各設定値を入力後、ウィンドウ下部にある[作成]をクリックします。

Table 12. 設定項目と設定値

設定項目	設定値
出力エイリアス	任意の名前 例) sqldatabase
シンク	SQLデータベース
インポートオプション	サブスクリプションからSQLデータベースを選択する
データベース	作成したデータベースを選択 例) sqlDb2018001
サーバー名	自動入力

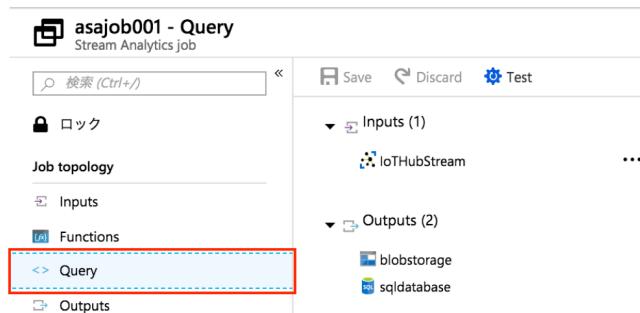
設定項目	設定値
ユーザー名	SQL Database作成時に指定した管理ユーザー 例) dbadmin
パスワード	SQL Database作成時に指定したパスワード 例) @dmin123
テーブル	tbl_averagedata



タスク6：Stream Analyticsのクエリ設定（SQLデータベースへの出力）

このタスクでは、Stream AnalyticsにSQLデータベースに出力するクエリを追加します。

1. Stream Analyticsの設定メニューにある[Query]をクリックします。



2. 右側のウィンドウ内のクエリを追加し、[保存]をクリックします。

WITH句の中でIoT

Hubにデータを送信したデバイス名を取得し(deviceId)、過去

1分間のデバイス毎の温度/湿度の平均値を取得してSQLデータベースに出力します。

```

WITH [averagedata] AS (
    SELECT
        IoTHubStream.IoTHub.ConnectionDeviceId as deviceId,
        ambient.temperature as temperature,
        ambient.humidity as humidity
    FROM
        [IoTHubStream]
)

SELECT
    deviceId,
    AVG(temperature) as temperature,
    AVG(humidity) as humidity,
    System.Timestamp as time
INTO
    [sqldatabase]
FROM
    [averagedata]
GROUP BY
    deviceId, TumblingWindow(minute, 1)

SELECT * INTO [blobstorage] FROM [IoTHubStream]

```

The screenshot shows the Azure Stream Analytics job configuration interface. On the left, there's a sidebar with options like 'Inputs', 'Outputs', and 'Query'. The 'Query' option is selected and highlighted with a blue background. In the main area, there's a 'Save' button with a red box around it, followed by 'Discard' and 'Test' buttons. Below these are sections for 'Inputs' (with 'IoTHubStream' listed) and 'Outputs' (with 'blobstorage' and 'sqldatabase' listed). To the right, the query code is displayed in a code editor window, with the first few lines of the WITH clause highlighted with a red box.

```

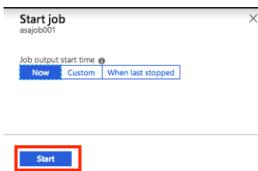
1 WITH [averagedata] AS (
2     SELECT
3         IoTHubStream.IoTHub.ConnectionDeviceId as deviceId,
4         ambient.temperature as temperature,
5         ambient.humidity as humidity
6     FROM
7         [IoTHubStream]
8 )

```

3. Stream Analyticsの設定メニューにある[Overview]をクリックし、ウィンドウ上部にある[Start]をクリックします。

The screenshot shows the 'Overview' page for the 'asajob001' Stream Analytics job. At the top, there's a search bar and three buttons: 'Start' (highlighted with a red box), 'Stop', and 'Delete'. Below the buttons, the status is shown as 'Created'. There are also links for 'アクティビティログ' (Activity Log) and 'アクセス制御 (IAM)' (Access Control).

4. [ジョブの開始]ウィンドウが表示されたら、ウィンドウ下部の[Start]をクリックします。



5. 通知ウィンドウで正常にStream Analyticsジョブが開始されることを確認します(数分かかります)。
6. SQLデータベースに格納されたデータを確認します。SQLデータベースの[クエリエディター]で、テーブルを選択し、[データを編集します]をクリックして1分間隔に集計されたデータを確認します。

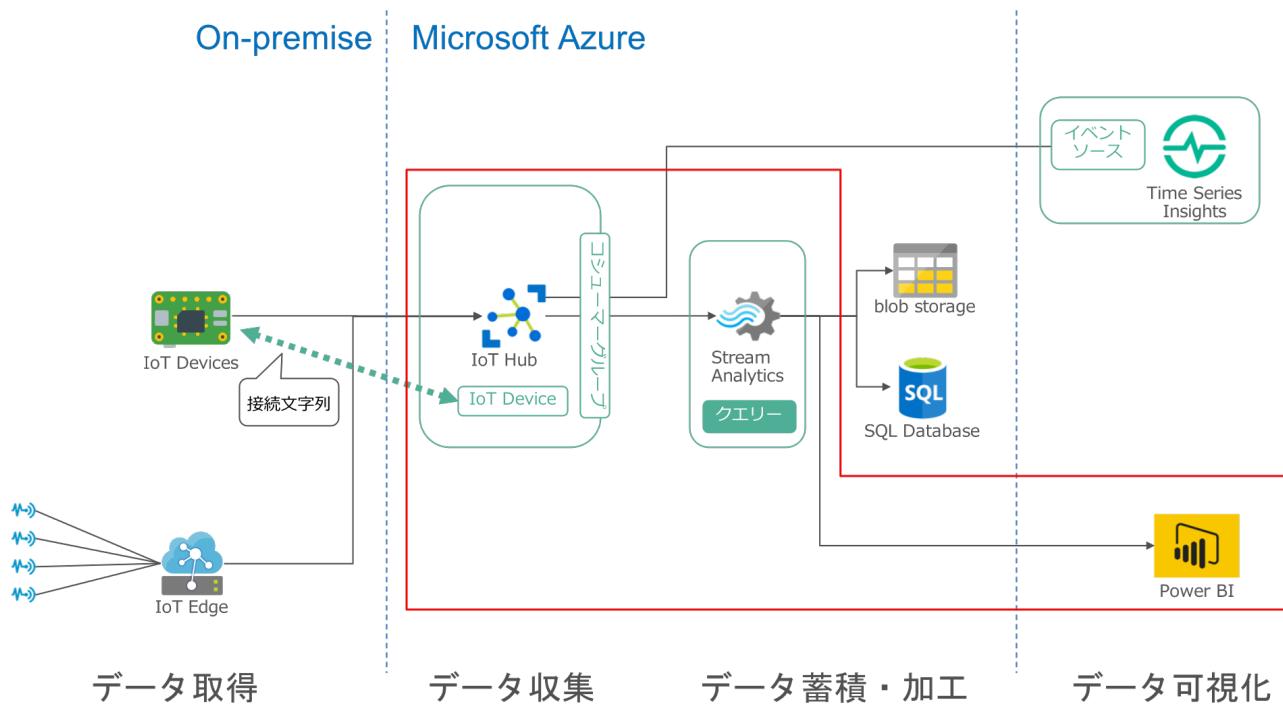
DEVICEID	TEMPERATURE	HUMIDITY	TIME
sampleddevice	25.9134954180056	69.6367632728572	2018-10-21T18:17:00.000000Z
sampleddevice	26.510661232806	70.2465003586438	2018-10-21T18:16:00.000000Z
sampleddevice	25.8709803057484	68.8401766695347	2018-10-21T18:18:00.000000Z
sampleddevice	26.7268852700579	69.9379833240598	2018-10-21T18:19:00.000000Z

[新しいクエリ]でSQLを直接入力して[実行]ボタンをクリックしてデータ一覧を確認することもできます。

```
select * from dbo.tbl_averagedata order by time
```

7. データを確認後、Stream Analyticsジョブを停止しておきます。



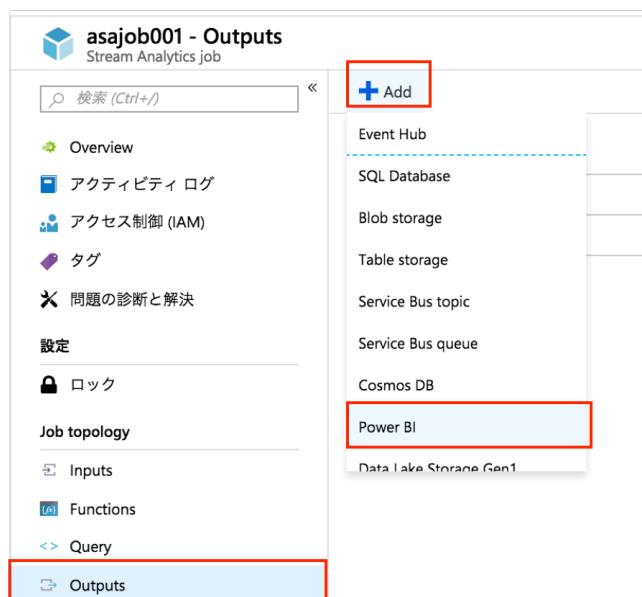


演習：Power BIでのデータ表示

この演習では、Stream AnalyticsからのストリーミングデータをPower BIでリアルタイム表示します。Stream AnalyticsにPower BI用の出力を追加し、Power BI側でストリーミングデータセットとして扱います。

タスク1：Stream Analyticsの出力設定（PowerBIへの出力）

1. Stream Analyticsを停止します。
2. Stream Analyticsの新規出力設定でpowerbiを追加します。



3. 設定項目を入力後、Power

BIの[Authorize]をクリックし、Power

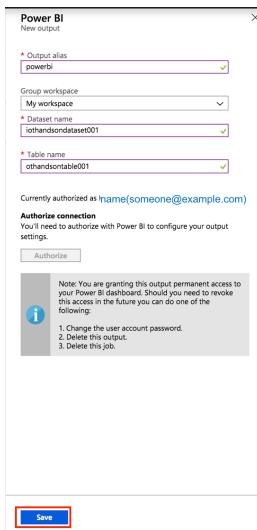
BIユーザーでサインインします。別のブラウザウィンドウが起動します。

Table 13. 設定項目と設定値

設定項目	設定値
出力エイリアス Output alias	任意の名前 例) powerbi
グループワークスペース Group workspace	My workspace
データセット名 Dataset name	任意の名前 例) dataset001
テーブル名 Table name	任意の名前 例) table001

The screenshot shows two windows side-by-side. On the left is the 'Power BI' configuration dialog titled 'New output'. It contains three input fields with red boxes around them: 'Output alias' (powerbi), 'Dataset name' (iothandsontdataset001), and 'Table name' (iothandsontable001). Below these is a section titled 'Authorize connection' with a note: 'You'll need to authorize with Power BI to configure your output settings.' A blue 'Authorize' button is highlighted with a red box. At the bottom, there's a note about granting permanent access to the Power BI dashboard with three options: changing the user account password, deleting the output, or deleting the job. On the right is a Microsoft Azure sign-in window titled 'Microsoft Azure' with the Microsoft logo. It shows the email 'someone@example.com' and a 'Next' button at the bottom.

4. Power BIの認証が完了したら、[Save]をクリックします。



TIP

下記URLに詳細説明の記載があります。 <https://docs.microsoft.com/ja-jp/azure/stream-analytics/stream-analytics-power-bi-dashboard>

5. Stream Analyticsのクエリの末尾にPower BI出力用のクエリを追加します。過去15秒間のデバイス毎の温度/湿度の平均値を取得してPower BIにプッシュで出力します。

```

WITH [averagedata] AS (
    SELECT
        IoTHubStream.IoTHub.ConnectionDeviceId as deviceId,
        ambient.temperature as temperature,
        ambient.humidity as humidity
    FROM
        [IoTHubStream]
)

SELECT
    deviceId,
    AVG(temperature) as temperature,
    AVG(humidity) as humidity,
    System.Timestamp as time
INTO
    [sqldatabase]
FROM
    [averagedata]
GROUP BY
    deviceId, TumblingWindow(minute, 1)

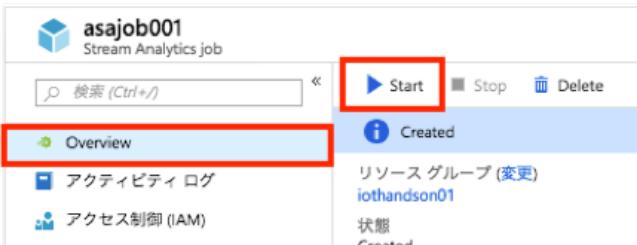
SELECT * INTO [blobstorage] FROM [IoTHubStream]

SELECT
    deviceId,
    AVG(temperature) as temperature,
    AVG(humidity) as humidity,
    System.Timestamp as time
INTO
    [powerbi]
FROM
    [averagedata]
GROUP BY
    deviceId, TumblingWindow(second, 15)

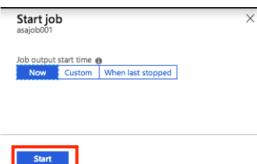
```

6. クエリを設定後、ジョブを開始します。

Stream AnalyticsのOverviewで[Start]をボタンをクリックします。



ダイアログで[Start]ボタンをクリックしてJobをスタートします。



7. Power BIにログインします。

<https://powerbi.microsoft.com/>
]をクリックしてサインインします。

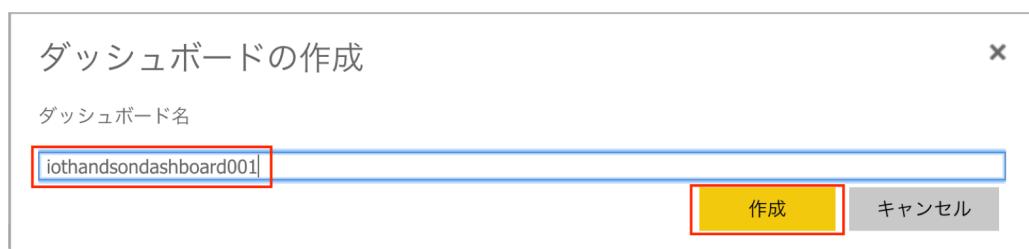
にアクセスし、上部の[サインイン]



8. サイドメニューからマイワークスペースを開き、ウィンドウ右上の[+作成]をクリックします。



9. [ダッシュボード]を選択し、任意の名前を設定します。 (例： dashboard001)



10. ウィンドウ上部の[+タイルの追加]をクリックし、リアルタイムデータ欄の[カスタムストリーミングデータ]を選択し、[次へ]をクリックします。



11. Stream Analyticsのデータセット名に設定したデータセット（例：dataset001）が、データセット欄に表示されていることを確認してください。表示されていればこれを選択し、[次へ]をクリックします。



CAUTION Stream Analyticsジョブが正常に動作していないと、データセットとして表示されません。

12. 視覚化タイプで[カード]を選択し、フィールドタブの[+値の追加]をクリックします。



13. プルダウンから”temperature”を選択します。



14. データラベルのタブを選択し、[小数点以下桁数の値]を設定します。(例: “2”)



15. ウィンドウ下部の[次へ]をクリックします。



16. [タイトル]、[サブタイトル]に任意の文字列をして、[適用]をクリックします。

タイトルの詳細

* 必須

詳細

タイトルとサブタイトルの表示

タイトル
温度

サブタイトル

既定値に戻す

技術的な詳細

戻る 適用 キャンセル

ダイアログ表示されますが、少し待っていると表示されなくなります。
すぐに非表示にしたい場合は、右上の×ボタンをクリックしてください。

CAUTION



17. 目的に応じて、ダッシュボードを完成させてください。

TIP

Power BI サービスのダッシュボードの詳細は、下記をご参照ください。
<https://powerbi.microsoft.com/ja-jp/documentation/powerbi-service-dashboards/>

18. タイルの視覚化タイプに”折れ線グラフ”を選択した場合

- a. [+タイトルの追加]をクリックし、タイトルの追加(ソースの選択)画面で[カスタムストリーミングデータ]を選択し、[次へ]をクリックします。
- b. カスタムストリーミングデータタイルの追加(ストリーミングデータセットの選択)画面でデータセットを選択し、[次へ]をクリックします。
- c. カスタムストリーミングデータタイルの追加(ストリーミングデータセットの選択>視覚化デザイン)画面で、折れ線グラフを選択しパラメータを入力します、

カスタム ストリーミング データ タイ ルの追加

ストリーミング データセットの選択 > 視覚化デザイン



視覚化タイプ
折れ線グラフ

軸
time

+ 値の追加

凡例
deviceid

+ 値の追加

値
temperature

+ 値の追加

表示する時間枠
最後 5 分

データセットの管理

戻る 次へ キャンセル

Table 14. 設定項目と設定値

設定項目	設定値
軸	time
凡例	deviceid
値	temperature
表示する時間枠	最後：5 分

□ データについて質問する

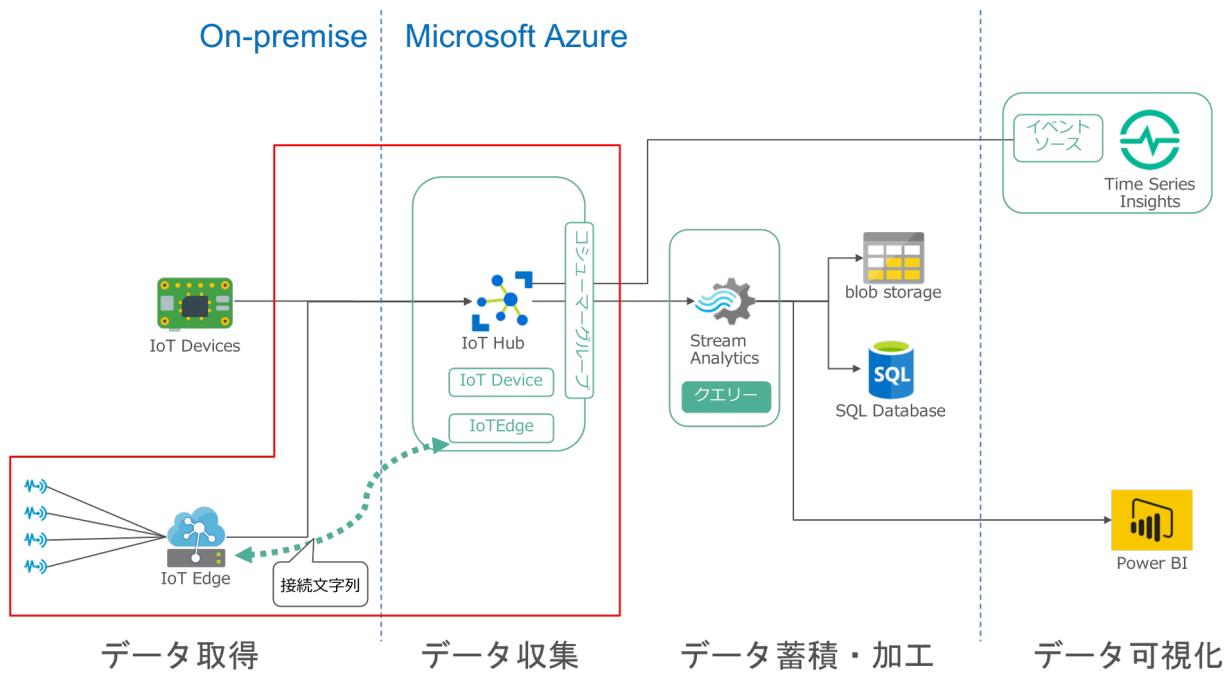
温度

26.16

temperature
基準 TIME, DEVICEID

deviceid ●sampledevice





演習: Azure IoT Edge デバイスの作成

この演習では、Azure上にデプロイしたVM(Ubuntu LTS 18.04)をAzure IoT Edgeデバイスとして作成し、シミュレーションデータ（温度/湿度）をIoT Hubに送信します。

タスク1：Linux VMのデプロイ

このタスクでは、IoT Edgeデバイスとして動作させる仮想マシン(Ubuntu LTS 18.04)をデプロイします。

1. Azure管理ポータル画面のサイドメニューから[リソースグループ]をクリックして、先に作成したリソースグループ(ex. iothandson)を選択します。
2. リソースグループの[追加]をクリックします。
3. "ubuntu server"で検索した結果から"Ubuntu LTS 18.04"を選択し、[作成]をクリックします。

The screenshot shows the Azure portal search results for "ubuntu server". The search bar at the top has "ubuntu server" entered. Below the search bar are three filter dropdowns: 料金 (All), Operating System (All), and 公開元 (All). The results section is titled "結果" (Results) and lists two items:

名前	公開元	カテゴリ
Ubuntu Server 16.04 LTS	Canonical	Compute
Ubuntu Server 18.04 LTS	Canonical	Compute

The second item, "Ubuntu Server 18.04 LTS", is highlighted with a red border.

4. [基本]で各種項目を設定します。

Table 15. 設定項目と設定値

設定項目	設定値
サブスクリプション	利用するサブスクリプションを選択
リソースグループ	[既存のものを使用]を選択し、先に作成した iothandson を選択
名前	任意のマシン名
場所	任意の場所を選択
可用性オプション	インフラストラクチャ冗長は必要ありません（デフォルトのまま）
イメージ	Ubuntu LTS 18.04
サイズ	（デフォルトのまま）
認証の種類	[パスワード]を選択
ユーザー名	任意のユーザー名（例：iotuser）
パスワード	次の条件を満たす、任意のパスワード (例：@dmin12345678) * 12-72文字 * 1つの小文字、1つの大文字、1つの数字および"¥"または"-"以外の1つの特殊文字
Azure Active Directoryでログインする（プレビュー）	オフ
パブリック受信ポート	選択したポートを許可
受信ポートを選択	SSH(22)

仮想マシンの作成

基本 ディスク ネットワーク 管理 ゲストの構成 タグ 確認および作成

LinuxまたはWindowsを実行する仮想マシンを作成します。Azure Marketplaceからイメージを選択するか、独自のカスタマイズされたイメージを使用します。[基本]タブについて [確認と作成]を完了させて既定のパラメータで仮想マシンをプロビジョニングするか、それぞれのタブを確認してフルカスタマイズを行います。

クラシックVMをお探しですか? Azure MarketplaceからVMを作成する

プロジェクトの詳細

デプロイされているリソースとコストを管理するサブスクリプションを選択します。フォルダのようなリソースグループを使用して、すべてのリソースを整理し、管理します。

- サブスクリプション:
- リソース グループ: iothandson [新規作成](#)

インスタンスの詳細

- 仮想マシン名: edgedevice
- 地域: 東日本
- 可用性オプション: インフラストラクチャ冗長は必要ありません
- イメージ: Ubuntu Server 18.04 LTS [すべてのイメージとディスクを参照する](#)
- サイズ: Standard D2s v3
2 vcpu 数、8 GB のメモリ [サイズを変更します](#)

管理者アカウント

認証の種類: パスワード SSH公開キー

- ユーザー名: iotuser
- パスワード:
- パスワードの確認:

Azure Active Directoryでログインする（プレビュー）: オフ オン

受信ポートの規則

パブリックインターネットからアクセスできる仮想マシンネットワークのポートを選択します。[ネットワーク]タブで、より限定的または細かくネットワークアクセスを指定できます。

- パブリック受信ポート: なし 選択したポートを許可する
- 受信ポートを選択: SSH

これらのポートは、インターネット上に公開されま
前回登録されたIPアドレスまたはドロップルールを使用して
変更される場合があります。詳細なドロップルールは後で更新
されます。

[確認および作成](#) [次: ディスク >](#)

5. [ディスク]で各種項目を設定します。

Table 16. 設定項目と設定値

設定項目	設定値
OSディスクの種類	HDDまたはSSD
アンマネージドディスクをしよう	いいえ

仮想マシンの作成

基本 ディスク ネットワーク 管理 ゲストの構成 タグ 確認および作成

Azure VM には、1つのオペレーティング システム ディスクと短期的なストレージの一時的ディスクがあります。追加のデータ ディスクをアタッチできます。VM のサイズによって、使用できるストレージの種類と、許可されるデータ ディスクの数が決まります。 詳細情報

ディスクのオプション

* OS ディスクの種類

選択した VM のサイズは Premium ディスクをサポートします。IOPS の高い作業負荷には Premium SSD がお勧めです。Premium SSD ディスクを搭載した仮想マシンは、99.9% の接続 SLA に適合します。

アンマネージド ディスクを ○ はい いいえ
使用

データ ディスク

仮想マシンに別のデータ ディスクを追加および構成したり、既存のディスクを接続したりすることができます。この VM には、一時ディスクも付属しています。

L... 名前 サイ... ディ... ホスト ...

[新しいディスクを作成し接続する](#) [既存のディスクの接続](#)

[確認および作成](#)

[前へ](#)

[次: ネットワーク >](#)

6. [管理]で各種項目を設定します。

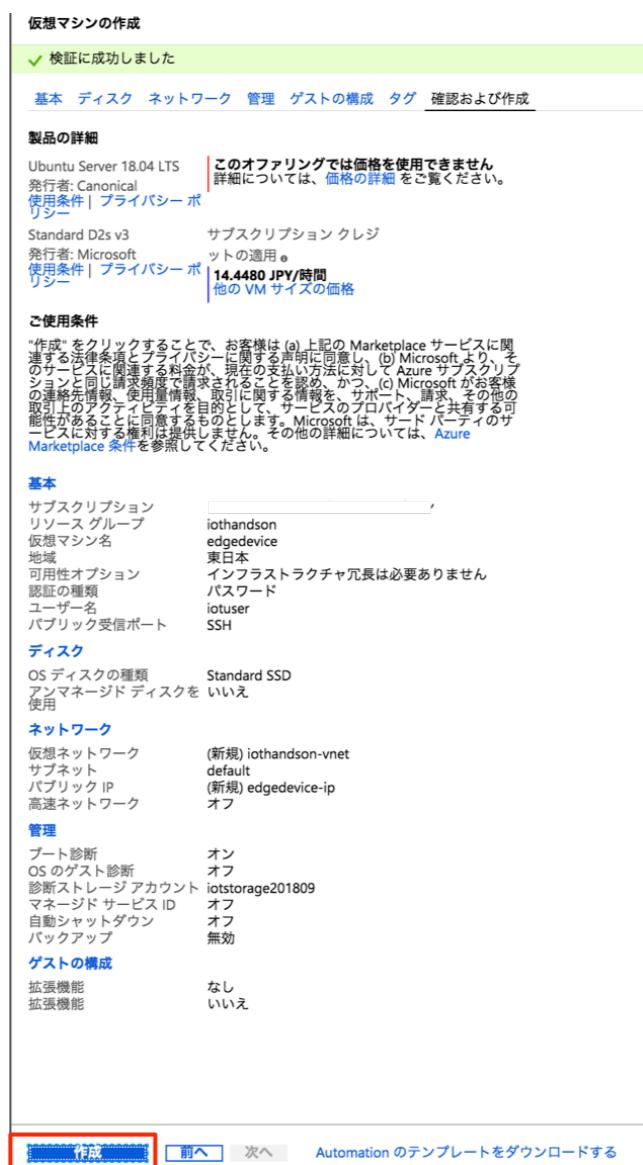
Table 17. 設定項目と設定値

設定項目	設定値
ブート診断	オン（デフォルトのまま）
OSのゲスト診断	オフ（デフォルトのまま）
診断ストレージアカウント	既存または新規作成
マネージドIDサービス	オフ（デフォルトのまま）
自動シャットダウンを有効	オフ（デフォルトのまま）
バックアップの有効化	オフ（デフォルトのまま）

image::images/iotedge-vmconfig-disk.png[+disk.png[pdfwidth='80%', width='80%']}

image::images/iotedge-vmconfig-

7. [概要]で[作成]ボタンをクリックします。



仮想マシンの作成

✓ 検証に成功しました

基本 ディスク ネットワーク 管理 ゲストの構成 タグ 確認および作成

製品の詳細

Ubuntu Server 18.04 LTS | このオファリングでは価格を使用できません。 詳細については、[価格の詳細](#)をご覧ください。
発行者: Canonical
使用条件 | プライバシー ポリシー

Standard D2s v3 | サブスクリプション クレジットの適用。
発行者: Microsoft
使用条件 | プライバシー ポリシー | 14,4480 JPY/時間
14,4480 JPY/時間 | 他の VM サイズの価格

ご使用条件

「作成」をクリックすることで、お客様は (a) 上記の Marketplace サービスに関する料金が、(b) Microsoft がお客様の払い方法に対応して Azure サブスクリプションと一緒に請求されることがあります。Microsoft がお客様の請求先情報を、サポート、請求、その他の取引上のアグリゲイト目的として、サービスのプロバイダーと共有する可能性があることに同意することになります。Microsoft は、サービスに対する権利は提供しません。その他の詳細については、[Azure Marketplace 条件](#)を参照してください。

基本

サブスクリプション	iothandson
リソース グループ	edgedevice
仮想マシン名	東日本
地域	インフラストラクチャ冗長は必要ありません
可用性オプション	パスワード
認証の種類	iotuser
ユーザー名	SSH
パブリック受信ポート	

ディスク

OS ディスクの種類	Standard SSD
アシマネージド ディスクを いいえ	使用

ネットワーク

仮想ネットワーク	(新規) iothandson-vnet
サブネット	default
パブリック IP	(新規) edgedevice-ip
高速ネットワーク	オフ

管理

ブート診断	オン
OS のゲスト診断	オフ
診断ストレージアカウント	iotstorage201809
マネージド サービス ID	オフ
自動シャットダウン	オフ
バックアップ	無効

ゲストの構成

拡張機能	なし
拡張機能	いいえ

作成

タスク2：コンテナランタイムのインストール

このタスクでは、IoT Edgeのランタイムをインストールします。

Edgeデバイス用にデプロイしたVMにIoT

1. 接続文字列を確認します。

VMのOverviewを開き[接続]をクリックし、右に表示されるダイアログで[SSH]タブを選択します。
Login using VM local accountに記載されている文字列をコピーします。



2. Cloud Shell またはSSHクライアントでVMに接続します。

前の手順でコピーした接続文字列をCloud Shellにペーストしてログインします。

```
ssh iotuser@[IPアドレス]
```

3. root ユーザにスイッチします

```
sudo -i
```

4. パッケージを取得するリポジトリを設定します。

```
curl https://packages.microsoft.com/config/ubuntu/16.04/prod.list > ./microsoft-prod.list
cp ./microsoft-prod.list /etc/apt/sources.list.d/
curl https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor >
microsoft.gpg
cp ./microsoft.gpg /etc/apt/trusted.gpg.d/
```

5. コンテナの実行エンジンのMobyをインストールします。

Moby(<http://mobyproject.org>) は、オープンソース版のDockerの実行環境です。

```
apt-get -y update  
apt-get -y install moby-engine  
apt-get -y install moby-cli
```

6. Moby が正常にインストールされたことを確認します。

```
docker run hello-world
```

以下のメッセージが表示されれば、mobyが正しくインストールされています。

```
Hello from Docker!  
This message shows that your installation appears to be working correctly.
```

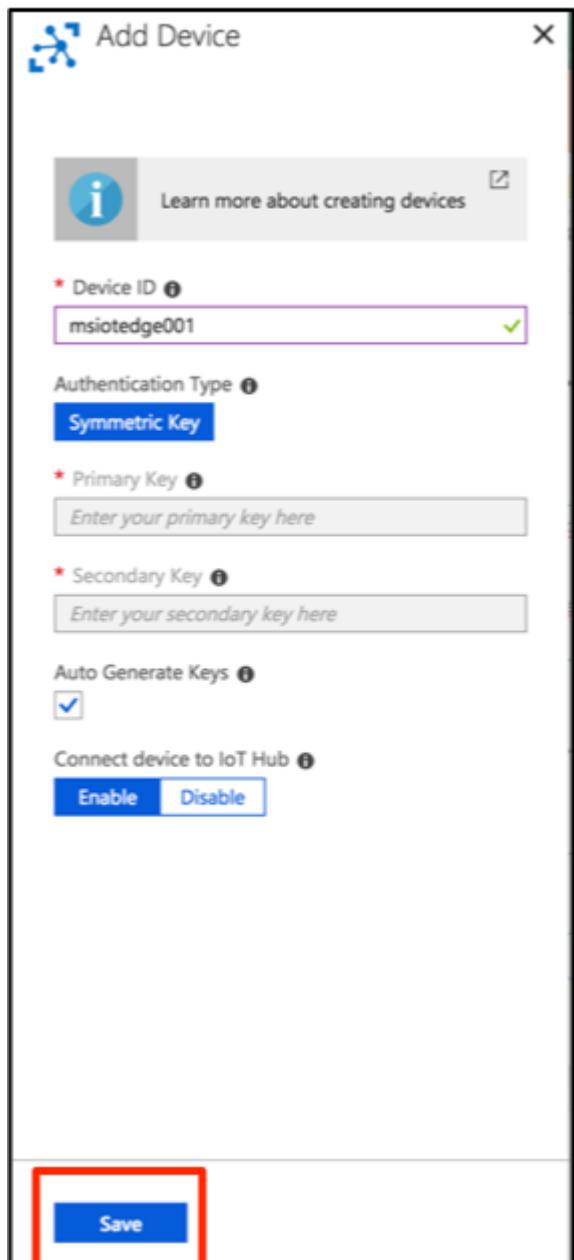
タスク3：IoT Edgeデバイスの追加

このタスクでは、IoT HubにIoT Edgeデバイスを新規登録します。

1. Azure管理ポータルで、先に作成したIoT Hub iothub-xx を開きます。
2. IoT Edgeをクリックして、[+ Add IoT Edge Device]をクリックします。

The screenshot shows the Azure IoT Hub management portal for 'iothub-201809'. On the left, there's a sidebar with sections like 'Automation スクリプト', 'Explorers' (Query explorer, IoT devices), 'Automatic Device Management' (IoT Edge, IoT device configuration), and 'Messaging' (File upload, Message routing). The 'IoT Edge' section is highlighted with a red box. On the right, the main area has a header with 'Add an IoT Edge device' (also highlighted with a red box) and other buttons for 'Add an IoT Edge deployment', 'Refresh', and 'Delete'. Below the header, there's a descriptive box about IoT Edge devices. Underneath, there's a table with columns for DEVICE ID, RUNTIME RESPON..., IOT EDGE MODUL..., CONNECTED CLIE..., and DEPLOYMENT CO... . A 'Run' button is also present.

3. Device IDを入力して[Save]ボタンをクリックします



4. IoT Edge Deviceを選択して接続文字列をコピーしておきます

IoT Edge Devices IoT Edge Deployments



IoT Edge Devices

IoT Edge devices have the IoT Edge runtime installed and are flagged as "IoT Edge device" in the device details. Each IoT Hub supports up to 1000 IoT Edge devices.

Query i
SELECT * FROM devices WHERE
optional (e.g. tags.location='US')

Execute

DEVICE ID	RUNTIME RESPONSE	MODULE COUNT	CONNECTED CLIENT ...	DEPLOYMENT COUNT
msiotedge001	N/A	0	0	0

Device Details
msiotedge001

Save Regenerate keys Device Twin Message To Device Direct Method Set Modules Refresh

Device Id i
msiotedge001

Primary key i

Secondary key i

Connection string—primary key i

Connection string—secondary key i

Connect device to IoT Hub i
Enable Disable

タスク4：IoT Edgeランタイムのインストール

このタスクでは、先にデプロイした Ubuntu VM に IoT Edgeランタイムをインストールします。

Ubuntu VMにSSHでログインして、以降のコマンドを順次実行します。

1. SSHでログイン

```
ssh iotuser@[IPアドレス]
```

2. rootユーザにスイッチします

```
sudo -i
```

3. IoT Edgeパッケージを取得します

```
apt-get -y install iotedge
```

4. IoT Edgeランタイムを構成します。

`/etc/iotedge/config.yaml` を編集して、デバイスの接続文字列を設定します。設定文字列の定義は26行目あたりにあります。

<ADD DEVICE CONNECTION STRING HERE>
を、前のタスクでコピーしておいた接続文字列で置き換えます。

編集前

```
provisioning:  
  source: "manual"  
  device_connection_string: "<ADD DEVICE CONNECTION STRING HERE>"
```

編集後

```
provisioning:  
  source: "manual"  
  device_connection_string: "HostName=xxxxx;DeviceId=xxx;SharedAccessKey=xxxxxxxx"
```

5. IoT Edgeランタイムを開始します

```
systemctl start iotedge
```

6. IoT Edgeランタイムのコンテナが動作していることを確認します

```
systemctl status iotedge
```

7. IoT Edgeランタイムのコンテナログを確認します

```
docker ps  
docker logs edgeAgent
```

`watch -n 10 docker ps` とすると、10秒間隔で`docker ps`コマンドを実行します。

8. Azure管理ポータルから、対象のIoT Edgeデバイスを開き[最新の情報に更新]をクリックします

9. デプロイされたモジュール一覧に `$edgeAgent` が追加されていることを確認します

NAME	TYPE	SPECIFIED IN DEPLOYMENT	REPORTED BY DEVICE	RUNTIME STATUS	EXIT CODE
<code>\$edgeAgent</code>	IoT Edge System module	No	Yes	running	-
<code>\$edgeHub</code>	Module Identity	N/A	N/A	N/A	N/A

タスク5：IoT Edgeデバイスにシミュレーションモジュールを追加

このタスクでは、IoT Edgeにマイクロソフトが今回するセンサーシミュレーションモジュールを追加します。センサーシミュレーションは、擬似的にテlemetryデータを生成した送信するモジュールです。

1. Azure管理ポータルから、対象のIoT Edgeデバイスを開き[Set Modules]をクリックします

2. Add Modulesの[+追加]-[IoT Edge Module]をクリックします

1 Add Modules (optional)

2 Specify Routes (optional)

3 Review Deployment

Container Registry Settings

NAME	ADDRESS	USER NAME	PASSWORD
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Deployment Modules

DESIRABLE STATUS	
+ Add	Delete
IoT Edge Module	running
Azure Stream Analytics Module	
Azure Machine Learning Module	

IoT Edge Modulesで項目を入力して[Save(保存)]ボタンをクリックします

設定項目	設定値
名前(name)	tempsensor
Image url	mcr.microsoft.com/azureiotedge-simulated-temperature-sensor:1.0

3. [Set Module]の画面に戻るので[Next]をクリックします

Set Modules

1 Add Modules (optional) 2 Specify Routes (optional) 3 Review Deployment

Container Registry Settings

NAME	ADDRESS	USER NAME	PASSWORD
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Deployment Modules

+ Add - Delete

<input type="checkbox"/> NAME	DESIRED STATUS
<input type="checkbox"/> tempsensor	running

[Configure advanced Edge Runtime settings](#)

Previous **Next** Submit

この画面では、メッセージのルーティングを定義します。デフォルトの "route": "FROM /messages/* INTO \$upstream" では、全てのモジュールの出力メッセージが \$upstream (IoT Hubを示す予約語) に送信されます。

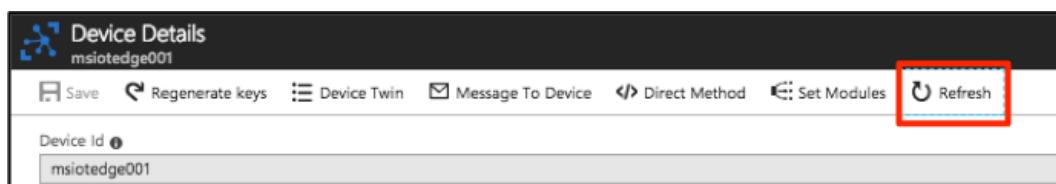
Set Modules

You can set routes between modules, which gives you the flexibility to send messages where they need to go without the need for additional services to process messages or to write additional code.

```
1 {
2   "routes": [
3     "route": "FROM /messages/* INTO $upstream"
4   ]
5 }
```

Previous **Next** Submit

4. 次の画面で[Submit]をクリックします
5. [Refresh]をクリックして \$edgeHub (IoT Edgeランタイムモジュールの一つ) と `tempsensor` モジュールが追加されていることを確認します



Connected

Modules IoT Edge Hub Connections Deployments

i This section specifies desired and reported modules. To adjust desired modules, click "Set Modules" or create a Deployment targeting the device. A delay may occur between the time a module is added or removed and the change is reported by the device. The "Reported by Device" column indicates whether a desired module has been reported by the device. The "Specified in Deployment" column indicates whether the module is specified in the applied deployment. A device may host up to ten modules.

NAME	SPECIFIED IN DEPL...	REPORTED BY DEVI...	RUNTIME STATUS	EXIT CODE	LAST START TIME (...	TYPE
\$edgeAgent	✓ Yes	✓ Yes	running	-	Tue Jul 17 2018 14:28:...	System module
\$edgeHub	✓ Yes	✓ Yes	running	-	Tue Jul 17 2018 15:06:...	System module
tempsensor	✓ Yes	✓ Yes	running	-	Tue Jul 17 2018 15:06:...	Custom module

tempsensorはシミュレーションデバイス、edgeHubはEdge Hub(IoT Edgeランタイム)を示しています

6. IoT EdgeがインストールされたUbuntu VMでDockerコンテナ一覧を確認します。

```
docker ps
```

7. IoT Edgeランタイム(Edge Hub)コンテナのログを確認します

```
docker logs edgeHub
```

8. シミュレーションデバイスコンテナのログから、送信メッセージを確認します。

```
docker logs tempsensor
```

シミュレーションデバイスで生成され、IoT Hubに送信されるデータの例

```
[  
  {  
    "machine": {  
      "temperature": 105.82414784646771,  
      "pressure": 10.663510514154549  
    },  
    "ambient": {  
      "temperature": 20.848158501250744,  
      "humidity": 25  
    },  
    "timeCreated": "2017-11-28T09:29:34.4441434Z"  
  }  
]
```

IoT Hubに送信されたメッセージは、Time Series Insightsなどで確認することができます。

NOTE

IoT EdgeにStream Analytics、Machine Learning、Funcsionsのモジュールを使用する場合は、以下のリンクをご参照ください（チュートリアルメニューの下に各モジュールの設定手順が掲載されています）
<https://docs.microsoft.com/ja-jp/azure/iot-edge/>

以上で、ハンズオンは終了です。お疲れ様でした！！