

#1 FPGA回路をリモートで書換 → 壮大なLチカハンズオン

C. Microsoft Azureの設定とIoT Edgeデバイスの登録

1. この章の内容とやること

この章では、今回利用するMicrosoft AzureのサービスであるContainer RegistryとIoT Hubの設定を行います。その後、DE10-NanoボードをIoT Edgeデバイスとして登録し、IoT Hubとの疎通を確認します。最後に、DE10-Nanoボードに出来合いのIoT Edgeモジュールを配置し、その動作を確認します。

本ハンズオンで使用するAzureサービスは、無料アカウントでは受講できません。利用料金の詳細については[G. 2. 今回のAzureサービスの利用料金](#)で解説します。

下記の項目の情報については、資料通り進める場合はこちらで指定した情報を入力いただきます。もちろん変更いただいて構いませんが、異なる情報を入力される場合は、ご自身で情報を適宜読み替えてください。

- リソースグループ名
- IoT Hub名
- IoT Edgeデバイス名
- IoT Edgeモジュール名

ということで、まずはWebブラウザでAzure Portalにサインインしておきましょう。

<https://portal.azure.com/>

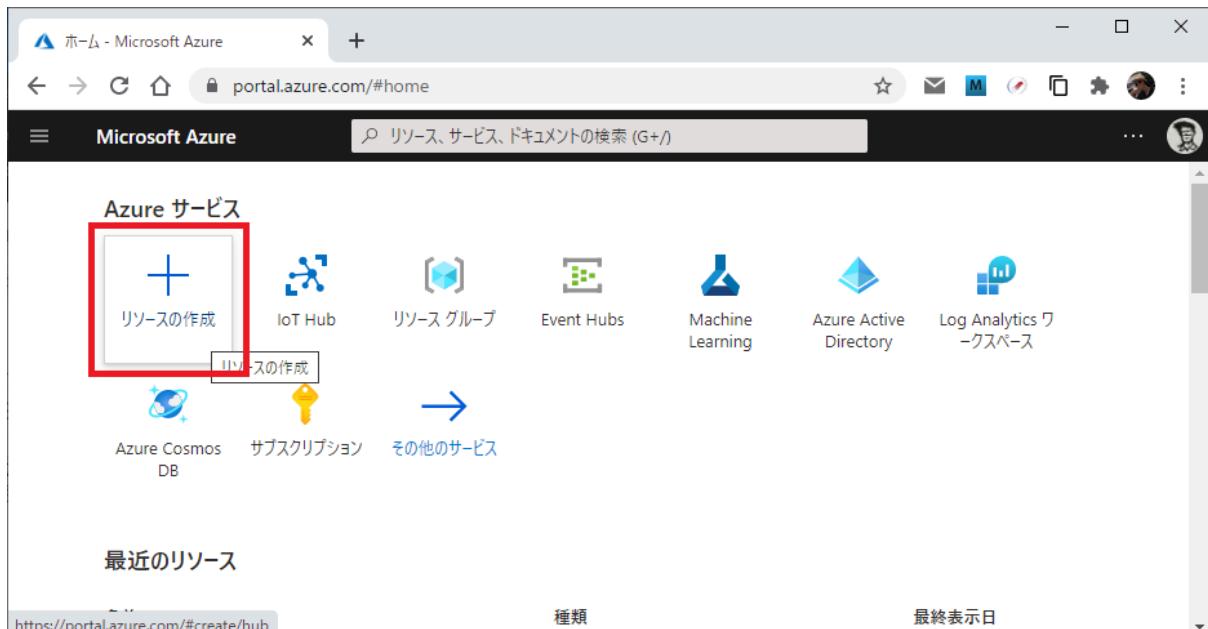
2. Container Registryの作成と設定

Azure Container Registry (ACR) とは、一言であらわすとAzure専用のコンテナイメージの格納庫です。Dockerを含む [OCI \(Open Container Initiative\)](#) の標準に準拠したコンテナを対象として、イメージの更新やコンテナのビルト、配置適用の自動化を支援します。

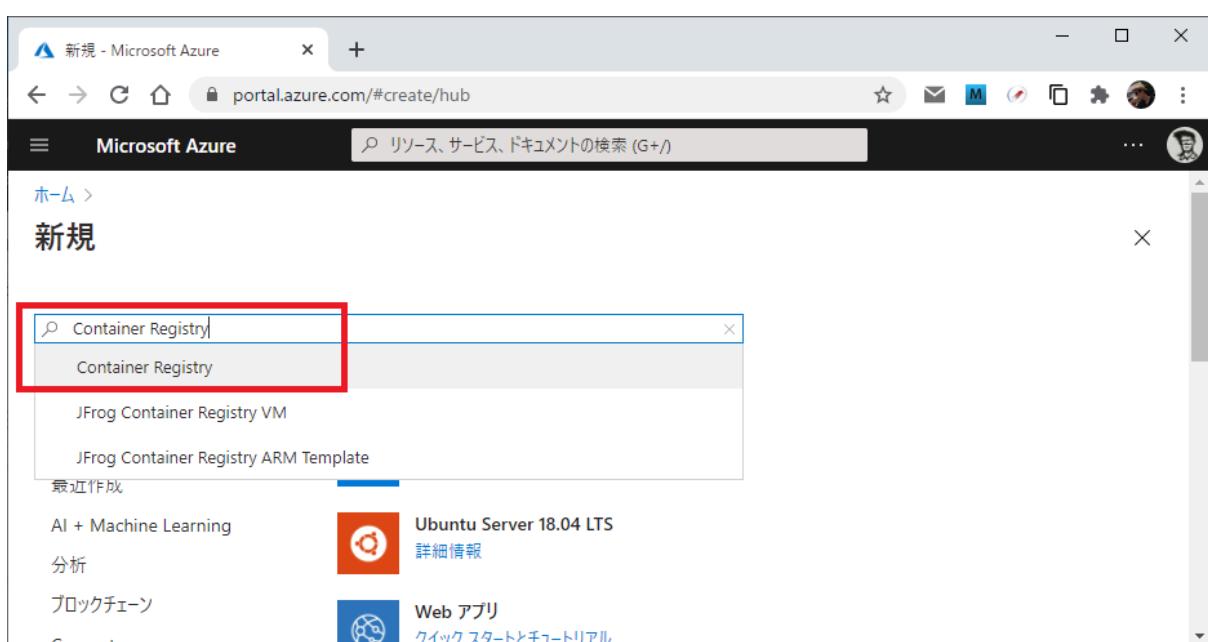
参考 : Azure Container Registry | Microsoft Azure

<https://azure.microsoft.com/ja-jp/services/container-registry/>

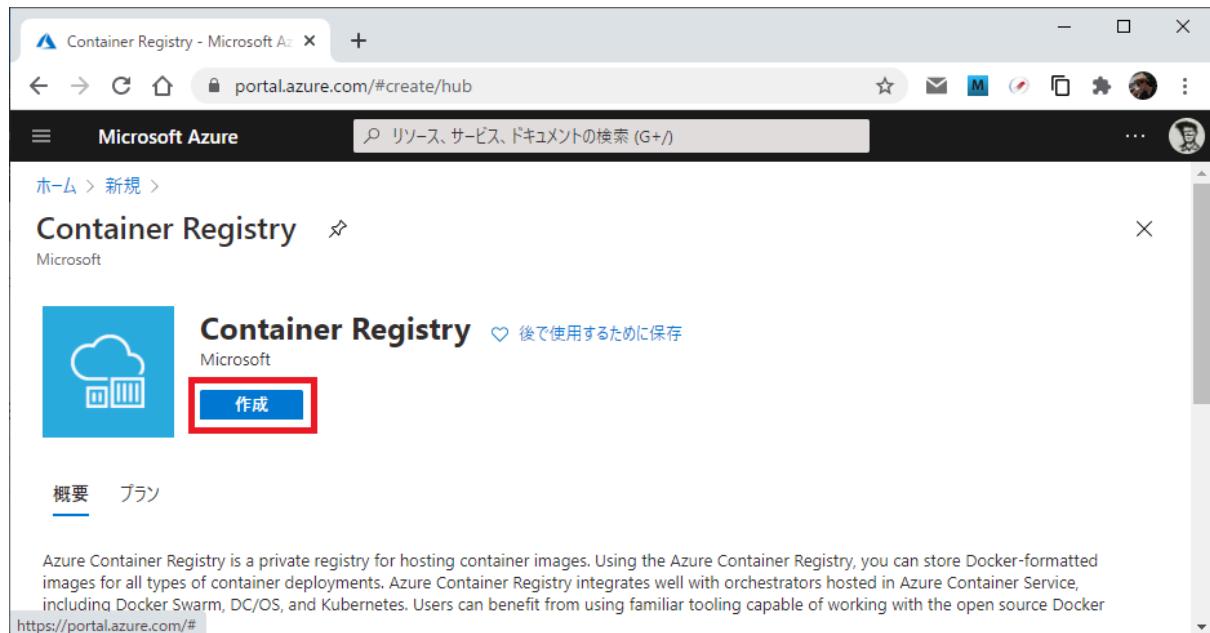
Azure PortalのTOPページにある **【+ リソースの作成】** をクリックします。



"Marketplace を検索" のフィールドに、 **Container Registry** を入力して、 Enterキーを押すか表示された候補をクリックします。



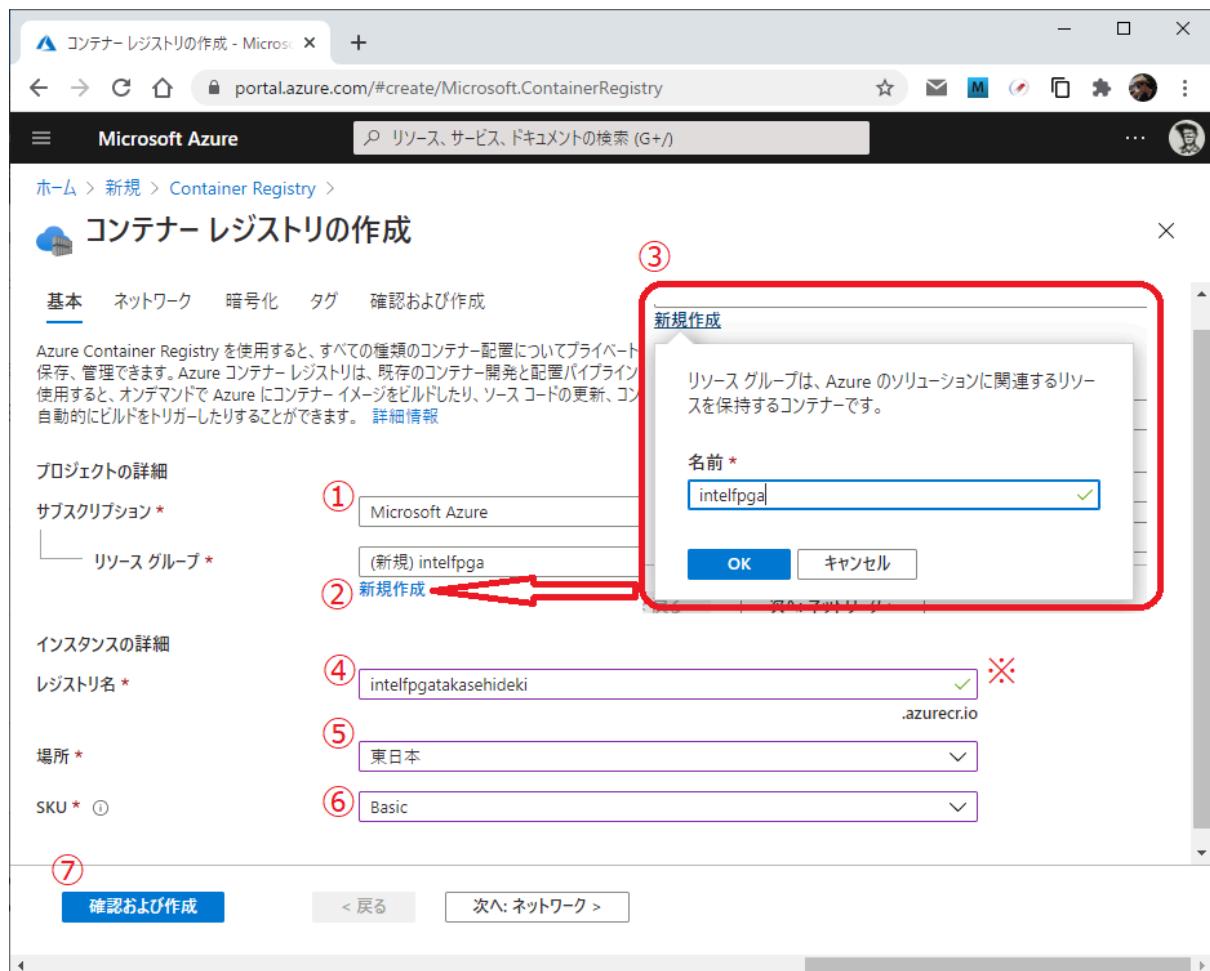
下記の画面になったら **【作成】** をクリックします。



「コンテナーレジストリの作成」画面になります。

[基本]の設定を行います。

- プロジェクトの詳細
 - ①【サブスクリプション】は、ご自身のものを選択してください。
例では **Microsoft Azure** を選択しています。
 - ②【リソースグループ】は、【新規作成】をクリックし、
○ ③【名前】に **intelfpga** を入力して **OK** をクリックしてください。
- インスタンスの詳細
 - ④【レジストリ名】は、**グローバルに一意のもの**を付ける必要があります。
5文字以上50字以下で、英字と数字のみ使用できます。
 - 例えば **intelfpgatakasehideki** などハンドル名を入れましょう。
 - ※のところが緑のチェックになっていることを確認してください。
 - ⑤【場所】は **東日本** を選択してください。
 - ⑥【SKU】は **Basic** を選択してください。
 - ACRの料金プランの選択です。今回はBasicで十分です。
- ⑦【確認および作成】をクリックします。



リソース作成のための設定情報を確認します。

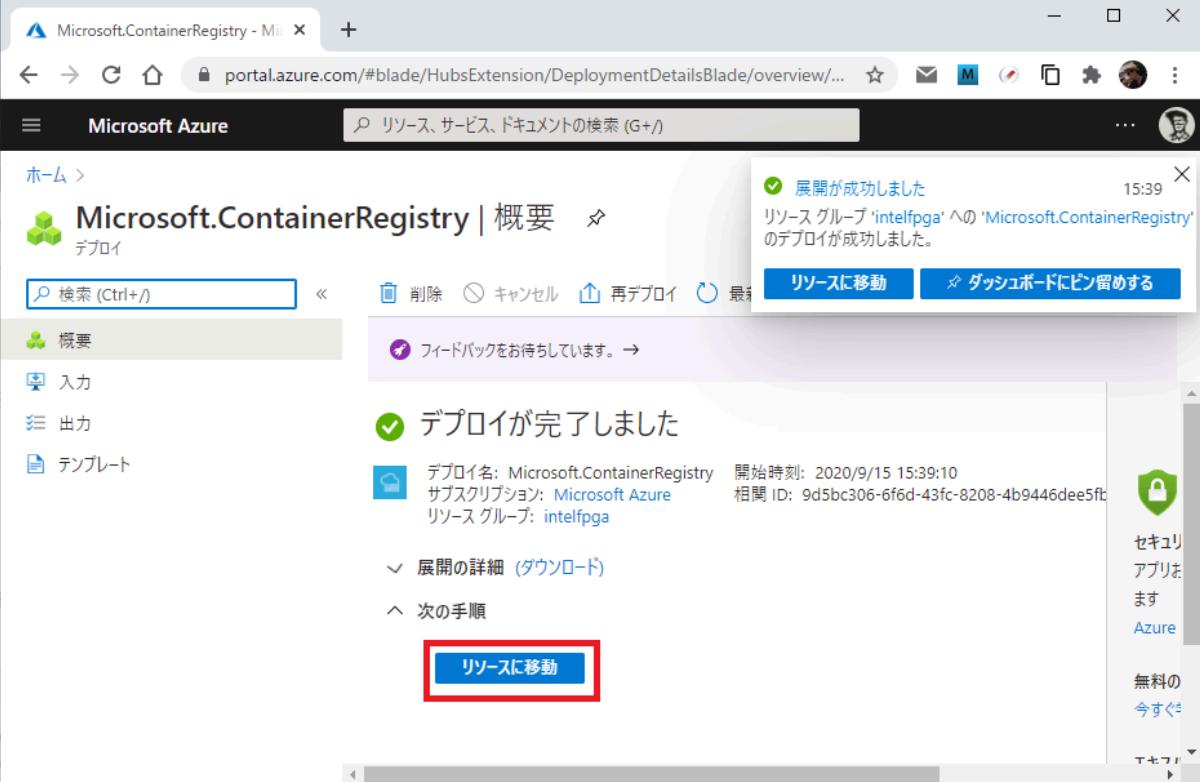
- 「検証に成功しました」と表示されることを確認してください。
- 【作成】をクリックします。



Container Registryのデプロイが進行します。少し待ちましょう。

下記の画面になったら、デプロイが完了しました。

- **【リソースに移動】**をクリックします。



Microsoft.ContainerRegistry - Microsoft Azure

リソース、サービス、ドキュメントの検索 (G+/)

Microsoft Azure

Microsoft.ContainerRegistry | 概要

デプロイ

検索 (Ctrl+ /)

概要

入力

出力

テンプレート

リソースに移動

ダッシュボードにピン留めする

展開が成功しました

リソース グループ 'intelfpga' への 'Microsoft.ContainerRegistry' のデプロイが成功しました。

デプロイ名: Microsoft.ContainerRegistry 開始時刻: 2020/9/15 15:39:10

サブスクリプション: Microsoft Azure 相関 ID: 9d5bc306-6f6d-43fc-8208-4b9446dee5fc

リソース グループ: intelfpga

展開の詳細 (ダウンロード)

次の手順

リソースに移動

展開されたリソースの設定情報を確認しておきましょう。

- ① レジストリ名: 例では **intelfpgatakasehideki**
- ② リソースグループ: **intelfpga**
- ③ サブスクリプション: **Microsoft Azure** など



intelfpgatakasehideki - Microsoft Azure

リソース、サービス、ドキュメントの検索 (G+/)

Microsoft Azure

intelfpgatakasehideki

コンテナー レジストリ

① 検索 (Ctrl+ /)

② リソース グループ (変更) intelfpga

③ サブスクリプション (変更) Microsoft Azure

移動 削除 更新

概要

アクティビティログ

アクセス制御 (IAM)

タグ

クリック スタート

イベント

リソース グループ (変更) intelfpga

場所 東日本

サブスクリプション (変更) Microsoft Azure

サブスクリプション ID

ログイン サーバー intelfpgatakasehideki.azurecr.io

作成日 2020/9/15 15:39 JST

SKU Basic

プロジェクトの状態

最後に、ACRの管理者ユーザーを有効にして、アクセスキーのパスワードを生成します。

- ① 左横メニューの「設定」>[アクセスキー]を開きます.
 - ②【管理者ユーザー】の【有効】をクリックして変更します.



Microsoft Azure | Microsoft Container Registry | intelfpgatakasehideki | アクセスキー

検索 (Ctrl+ /)

概要

レジストリ名: intelfpgatakasehideki

ログインサーバー: intelfpgatakasehideki.azurecr.io

② 管理者ユーザー ①

有効 (Red box)

無効

① アクセスキー

② 暗号化

③ ID

下の画面のようにパスワードが表示されたらOKです。



intelfpgatakasehideki | アクセスキー

レジストリ名
intelfpgatakasehideki

ログインサーバー
intelfpgatakasehideki.azurecr.io

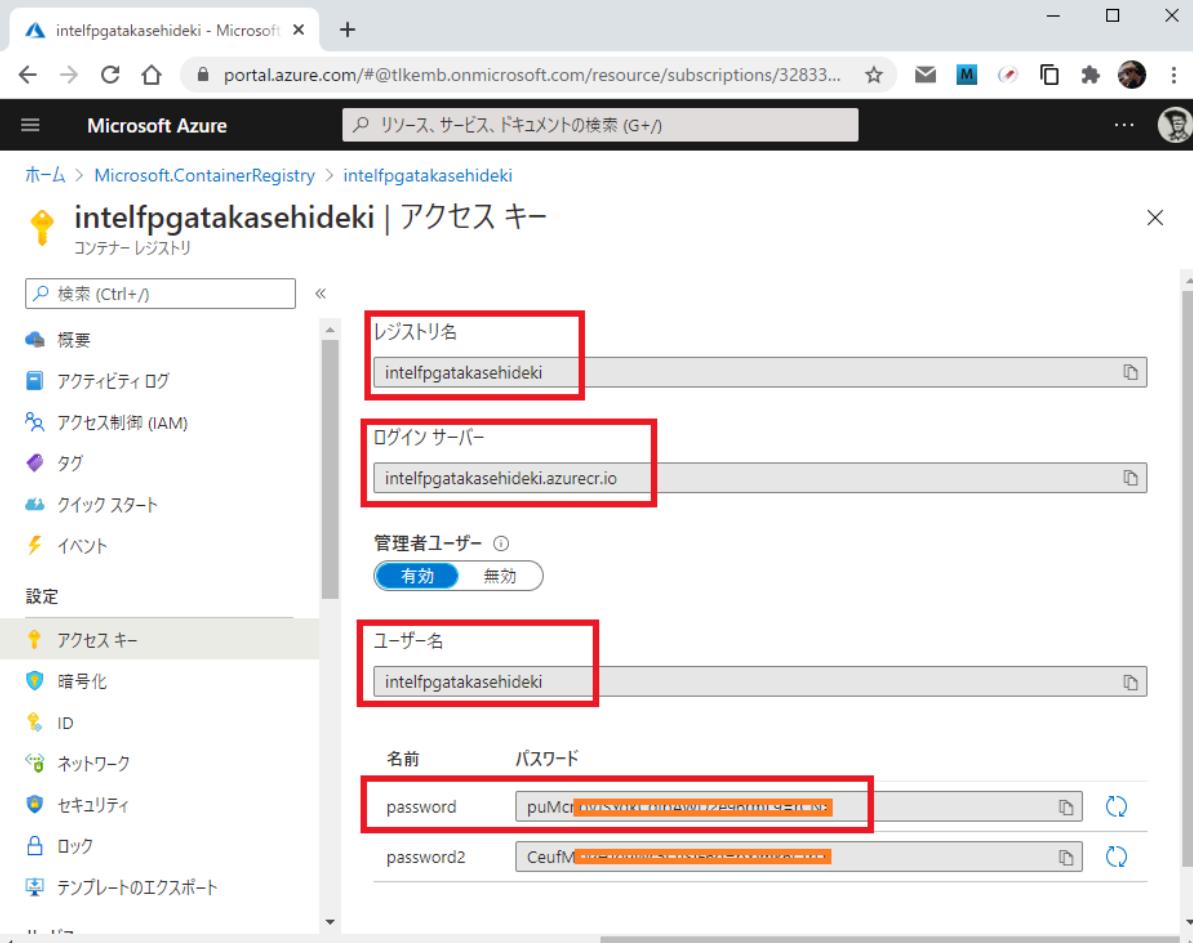
管理者ユーザー ①
有効 無効

ユーザー名
intelfpgatakasehideki

名前	パスワード
password	puMcXXXXXXXXXXXXXX
password2	CeufMXXXXXXXXXXXXXX

ここに表示されている次の情報は、VS Codeで開発用のプロジェクトを作成するときに必要となります。メモ帳などにコピペして控えておくと良いでしょう。

- 「レジストリ名」
- 「ログイン サーバー」
- 「ユーザー名」
- 「password」 ※password2は不要です



The screenshot shows the Microsoft Azure portal with the URL portal.azure.com/#@tlkemb.onmicrosoft.com/resource/subscriptions/32833.... The page displays the 'Access Keys' for a Container Registry named 'intelfpgatakasehideki'. The 'Registry Name' is listed as 'intelfpgatakasehideki', the 'Login Server' is 'intelfpgatakasehideki.azurecr.io', and the 'User Name' is 'intelfpgatakasehideki'. The 'password' field is highlighted with a red box. The left sidebar shows various settings like 'Overview', 'Logs', 'IAM', 'Tags', 'Quick Start', and 'Events'. The 'Access Keys' section is currently selected.

ここまでのお作業に関するMicrosoftの公式ドキュメントは、下記をご参照ください。

参考: クイックスタート - ポータルでのレジストリの作成 - Azure Container Registry | Microsoft Docs

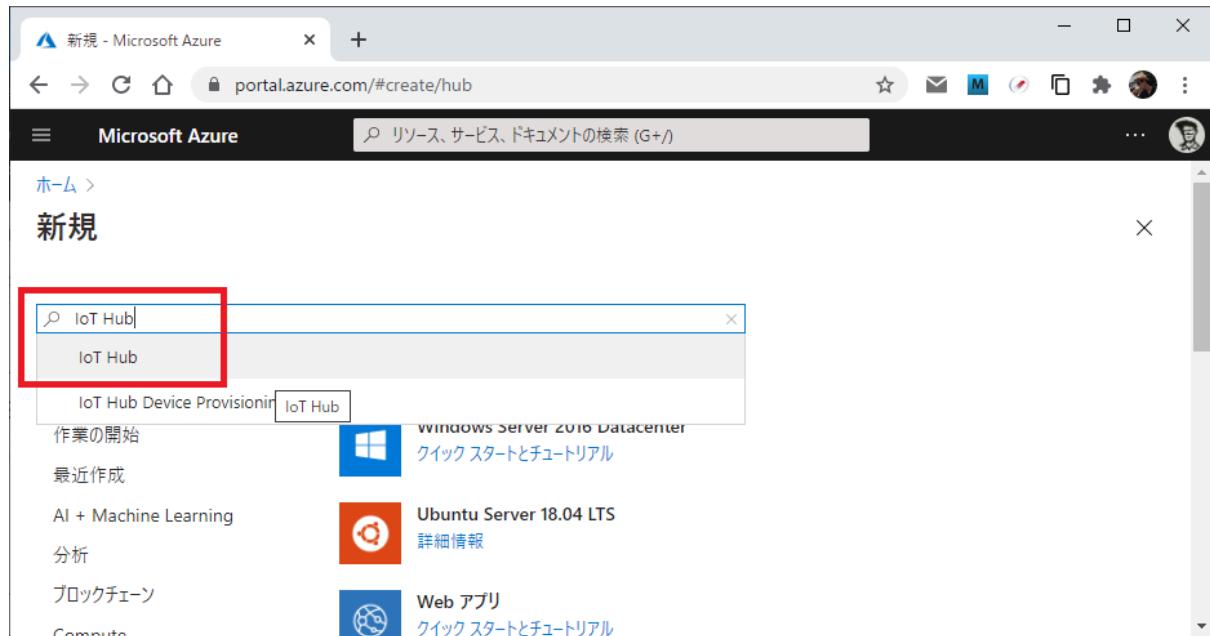
<https://docs.microsoft.com/ja-jp/azure/container-registry/container-registry-get-started-portal>

3. IoT Hubの作成

次はIoT Hubを作成します。

Azure PortalのTOPページに戻って 【+ リソースの作成】 をクリックします。

"Marketplace を検索" のフィールドに、 IoT Hub を入力して、 Enterキーを押すか表示された候補をクリックします。



下記の画面になったら 【作成】 をクリックします。



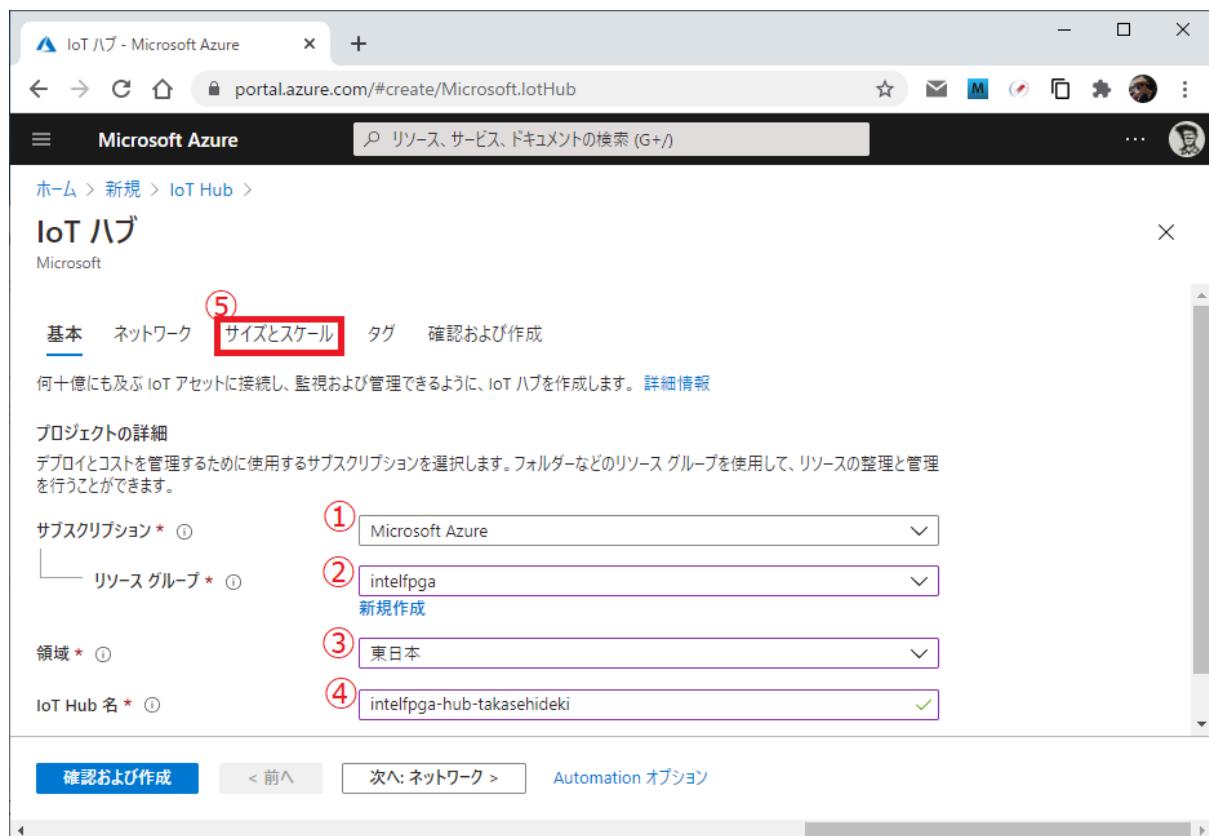
「IoT ハブ」画面になります。

[基本]の設定を行います。

- プロジェクトの詳細

- ①【サブスクリプション】は、ご自身のものを選択してください。
例では **Microsoft Azure** を選択しています。
 - ②【リソースグループ】は、先ほど作成した **intelfpga** を選択します。
 - ③【領域】は、**東日本** を選択してください。
 - ④【IoT Hub 名】は、**グローバルに一意のもの**を付ける必要があります。
英数字とハイフンが使えます。
 - 例えば **intelfpga-hub-takasehideki** などハンドル名を入れましょう
 - ※のところが緑のチェックになっていることを確認してください。
- 次は[サイズとスケール]を設定します。⑤のところをクリックしてください。

- まだ 確認および作成 はクリックしません。



IoT ハブ - Microsoft Azure

portal.azure.com/#create/Microsoft.IotHub

Microsoft Azure

IoT ハブ

基本 ネットワーク サイズとスケール タグ 確認および作成

何十億にも及ぶ IoT アセットに接続し、監視および管理できるように、IoT ハブを作成します。詳細情報

プロジェクトの詳細

デプロイとコストを管理するために使用するサブスクリプションを選択します。フォルダーなどのリソース グループを使用して、リソースの整理と管理を行うことができます。

サブスクリプション * ① Microsoft Azure

リソース グループ * ① intelfpga
新規作成

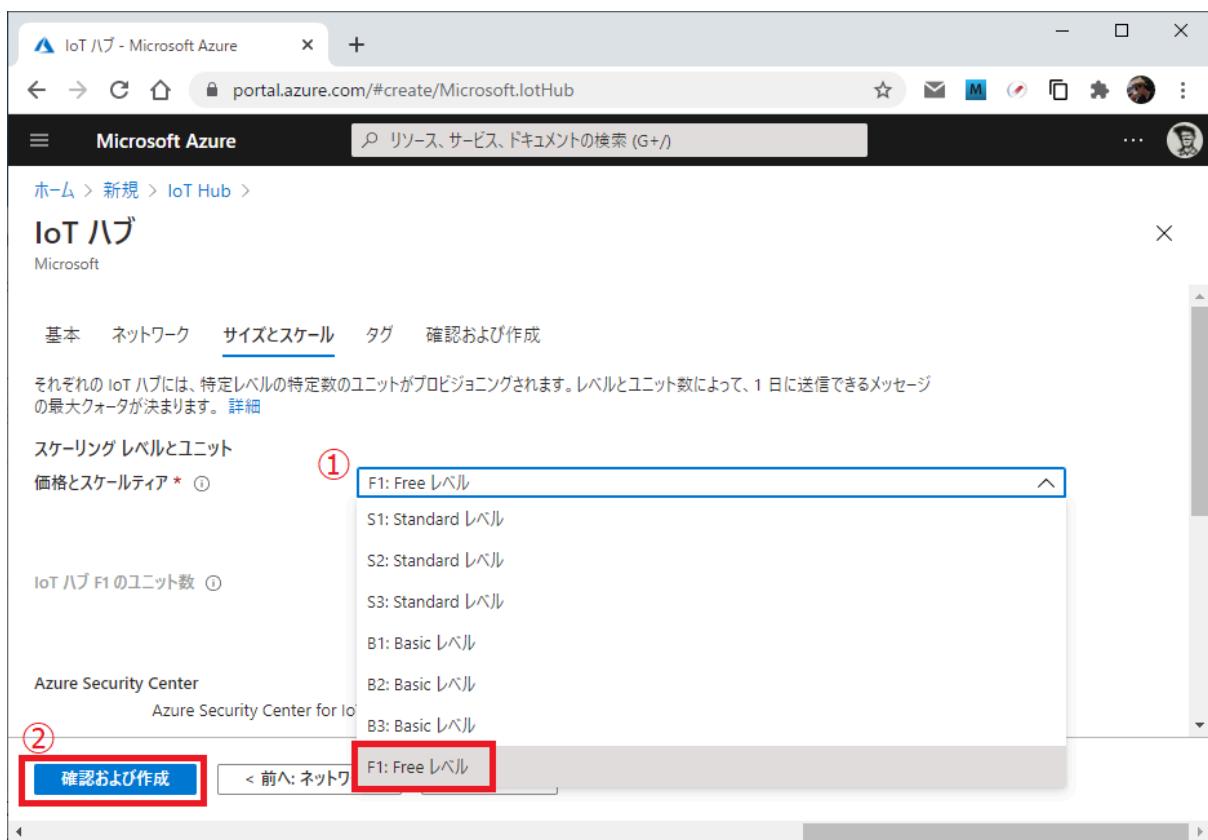
領域 * ① 東日本

IoT Hub 名 * ① intelfpga-hub-takasehideki

確認および作成 < 前へ 次へ: ネットワーク > Automation オプション

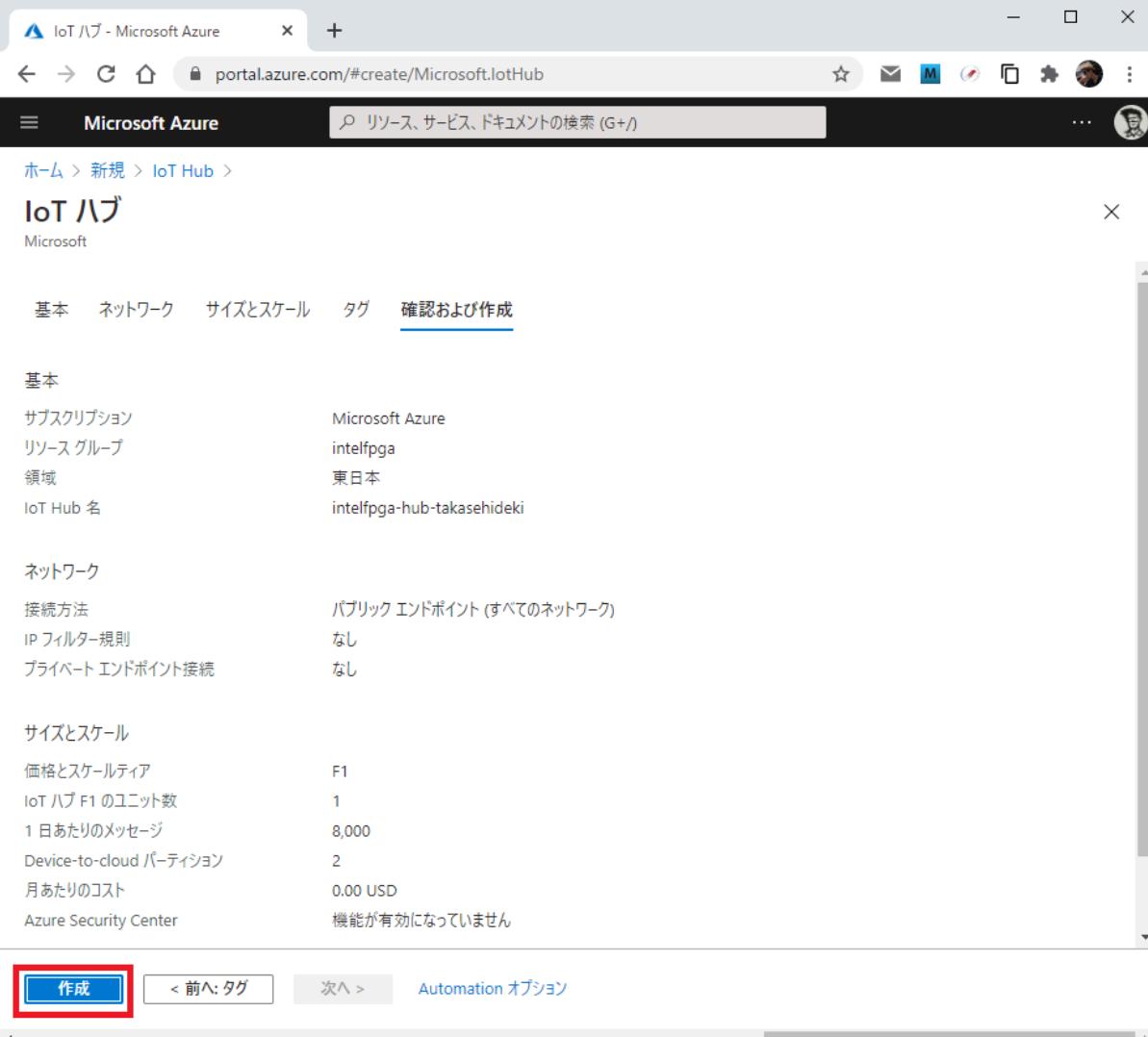
[サイズとスケール]の設定を行います。

- スケーリング レベルとユニット
 - ①【価格とスケールティア】は、**F1: Freeレベル** を選択してください。
 - F1は無料ですが、サブスクリプションごとに1つしか作成することができません。赤字のエラーメッセージが表示される場合は、既存のF1のIoT Hubを削除するか **S1: Standardレベル** を選択してください。
 - ここでF1を選択して作成したIoT Hubは、あとから有料のレベルに変更することはできません。
 - 参考: Azure IoT Hub クォータと調整について | Microsoft Docs
<https://docs.microsoft.com/ja-jp/azure/iot-hub/iot-hub-devguide-quotas-throttling>
 - ②【確認および作成】をクリックします。



リソース作成のための設定情報を確認します。

- **【作成】** をクリックします。



IoT ハブ - Microsoft Azure

portal.azure.com/#create/Microsoft.IotHub

Microsoft Azure リソース、サービス、ドキュメントの検索 (G+/)

IoT ハブ

Microsoft

ホーム > 新規 > IoT Hub >

IoT ハブ

確認および作成

基本

サブスクリプション Microsoft Azure
リソース グループ intelfpga
領域 東日本
IoT Hub 名 intelfpga-hub-takasehideki

ネットワーク

接続方法 パブリック エンドポイント (すべてのネットワーク)
IP フィルター規則 なし
プライベート エンドポイント接続 なし

サイズとスケール

価格とスケールティア F1
IoT ハブ F1 のユニット数 1
1 日あたりのメッセージ 8,000
Device-to-cloud パーティション 2
月あたりのコスト 0.00 USD
Azure Security Center 機能が有効になっていません

作成 < 前へ: タグ 次へ > Automation オプション

IoT Hubのデプロイが進行します。少し待ちましょう。

下記の画面になったら、デプロイが完了しました。

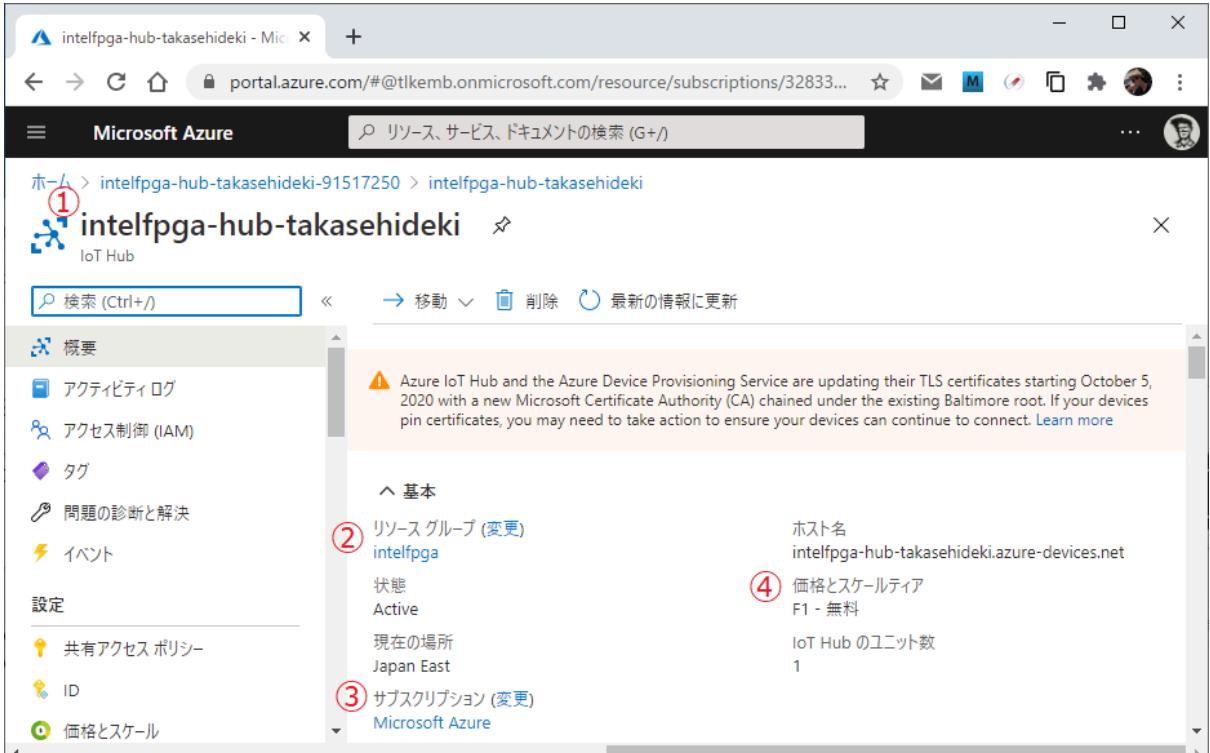
- **【リソースに移動】**をクリックします。



The screenshot shows the Microsoft Azure portal interface. The title bar reads "intelfpga-hub-takasehideki-91517250 | 概要". The main content area displays a green checkmark icon and the text "デプロイが完了しました" (Deployment completed). Below this, deployment details are listed: "デプロイ名: intelfpga-hub-takasehideki-91517250", "開始時刻: 2020/9/15 17:02:51", "サブスクリプション: Microsoft Azure", and "リソース グループ: intelfpga". There are also links for "展開の詳細 (ダウンロード)" and "次の手順". A prominent blue button at the bottom is labeled "リソースに移動". The left sidebar shows navigation links for "概要", "入力", "出力", and "テンプレート". The top navigation bar includes a search bar and various Azure service icons.

展開されたリソースの設定情報を確認しておきましょう。

- ① IoT Hub名: 例では **intelfpga-hub-takasehideki**
- ② リソースグループ: **intelfpga**
- ③ サブスクリプション: **Microsoft Azure**
- ④ 価格とスケールティア: **F1 - 無料**
など



The screenshot shows the Microsoft Azure portal interface for an IoT Hub. The URL in the address bar is <https://portal.azure.com/#@tlkemb.onmicrosoft.com/resource/subscriptions/32833...>. The page title is "intelfpga-hub-takasehideki - Microsoft Azure". The main content area shows the IoT Hub configuration with the following details:

- ① リソース名:** intelfpga-hub-takasehideki
- ② リソースグループ:** intelfpga
- ③ サブスクリプション:** Microsoft Azure
- ④ 価格とスケールティア:** F1 - 無料

Additional information visible on the page includes:

- A warning message: "Azure IoT Hub and the Azure Device Provisioning Service are updating their TLS certificates starting October 5, 2020 with a new Microsoft Certificate Authority (CA) chained under the existing Baltimore root. If your devices pin certificates, you may need to take action to ensure your devices can continue to connect. [Learn more](#)"
- ホスト名:** intelfpga-hub-takasehideki.azure-devices.net
- 状態:** Active
- 現在の場所:** Japan East
- IoT Hub のユニット数:** 1

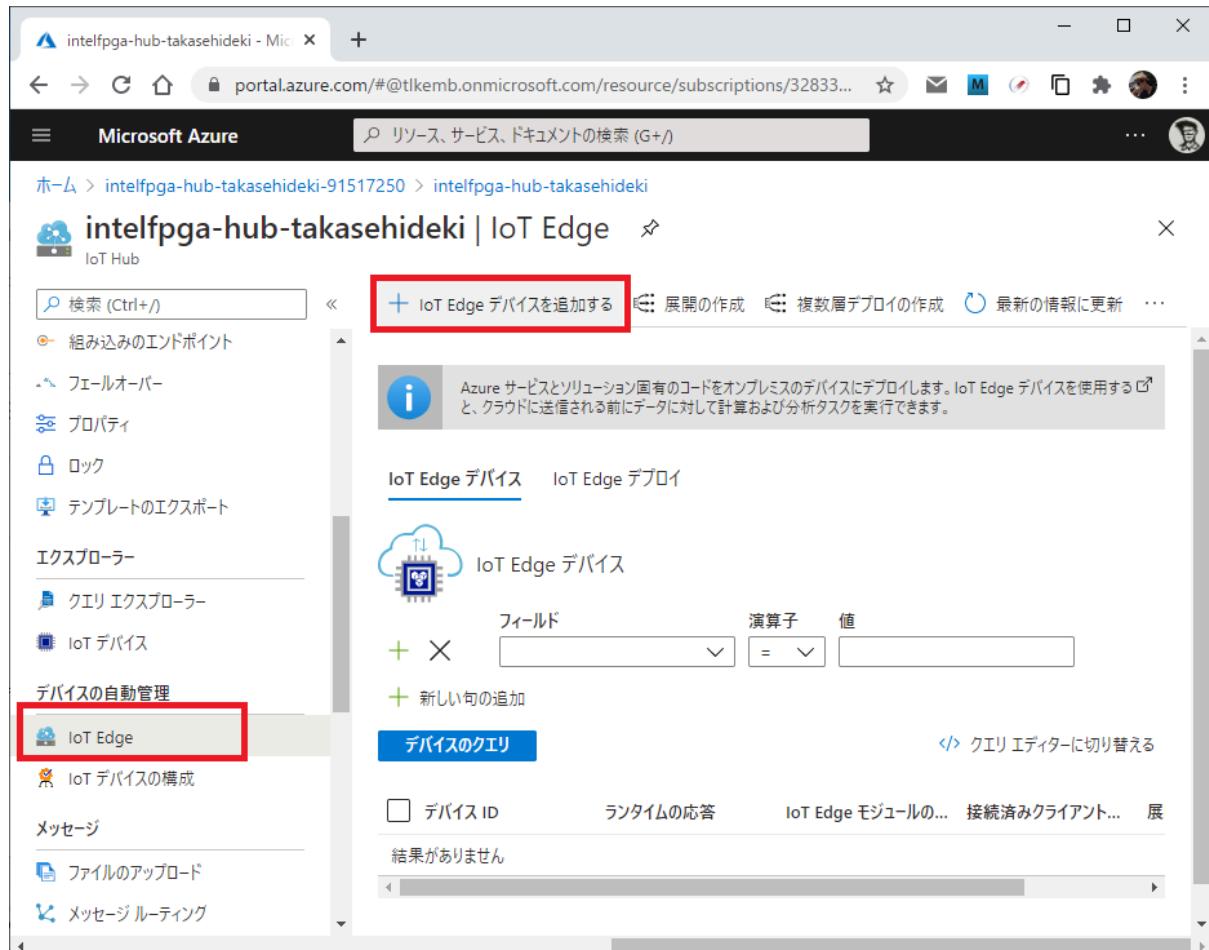
IoT Hubの作成は以上です。

4. IoT Edgeデバイスの作成

次はIoT Hub上にIoT Edgeデバイスを追加します。

先ほど作成したIoT Hubのリソースのページ(例では [intelfpga-hub-takasehideki](#))に移動して、左横メニューの「デバイスの自動管理」>[IoT Edge]を開きます。

【+ IoT Edge デバイスを追加する】をクリックします。



The screenshot shows the Microsoft Azure portal interface for managing IoT Edge devices. The main title is 'intelfpga-hub-takasehideki | IoT Edge'. On the left, a sidebar menu is open, showing 'IoT Edge' selected under 'デバイスの自動管理'. The main content area has a heading 'IoT Edge デバイス' and a sub-section 'IoT Edge デバイス'. Below this is a table with columns 'フィールド', '演算子', and '値'. A button '+ 新しい句の追加' is visible. At the bottom, there is a section 'デバイスのクエリ' with a table showing results. A red box highlights the 'IoT Edge デバイスを追加する' button at the top right of the main content area.

「デバイスの作成」画面になります。

- 【デバイス ID】に **de10-nano** と入力します。任意のものでも構いません。
- 他の設定はデフォルトのままで構いません。
- 【保存】**をクリックします。



デバイスの作成 - Microsoft Azure

Microsoft Azure リソース、サービス、ドキュメントの検索 (G+/)

ホーム > intelfpga-hub-takasehideki-91517250 > intelfpga-hub-takasehideki > デバイスの作成

Azure IoT 用に認定されたデバイスをデバイス カタログで検索します

デバイス ID * ①
de10-nano ✓

認証の種類 ①
対称キー X.509 自己署名済み

主キー ①
プライマリ キーを入力してください

セカンダリ キー ①
セカンダリ キーを入力してください

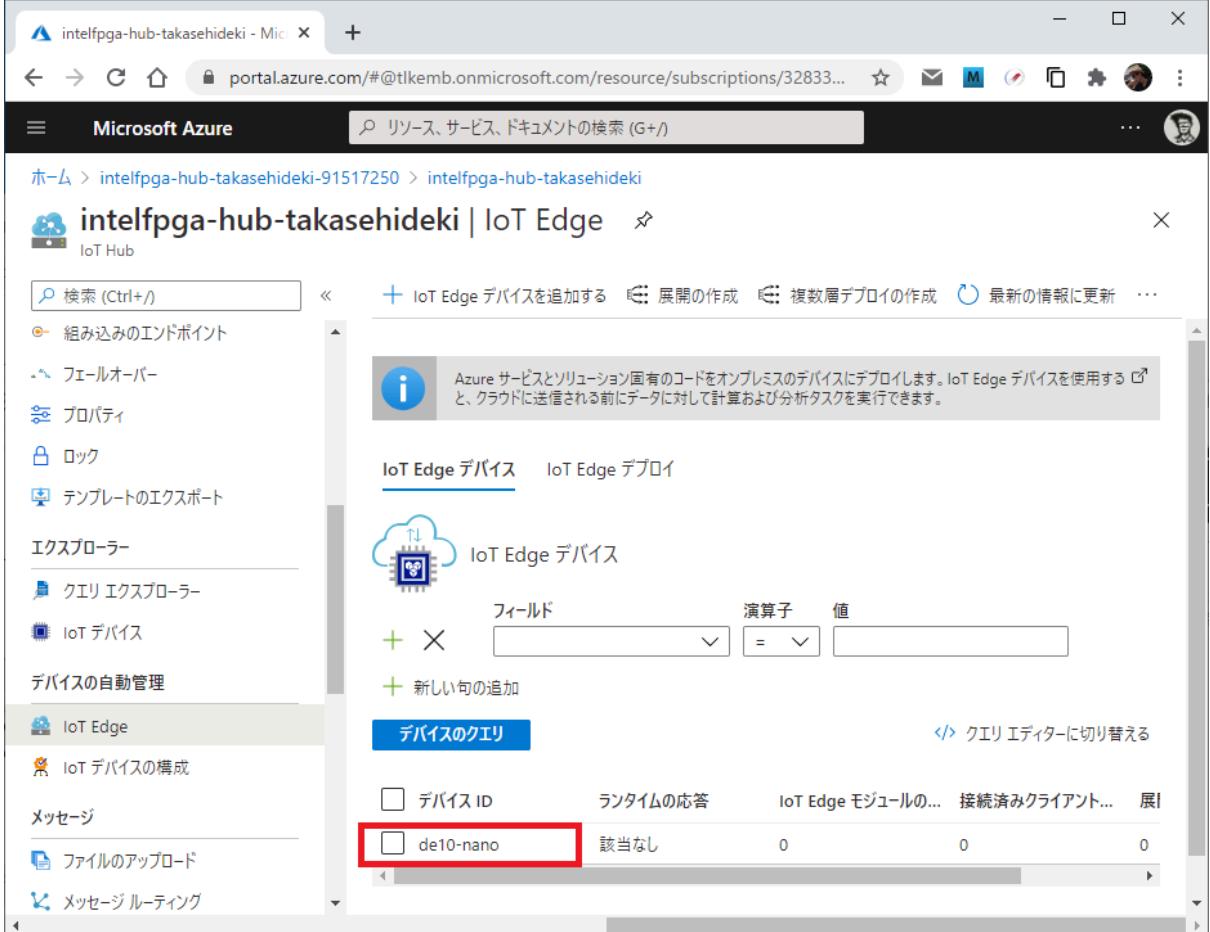
自動生成キー ①
✓

このデバイスを IoT ハブに接続する ①
有効化 無効化

子デバイス ①
0
子デバイスの選択

保存

ページが下記のように戻り、追加したデバイス IDである **de10-nano** が表示されていることを確認します。ここをクリックして開きます。
(このページは左横メニューの「デバイスの自動管理」>[IoT Edge]で開けます)

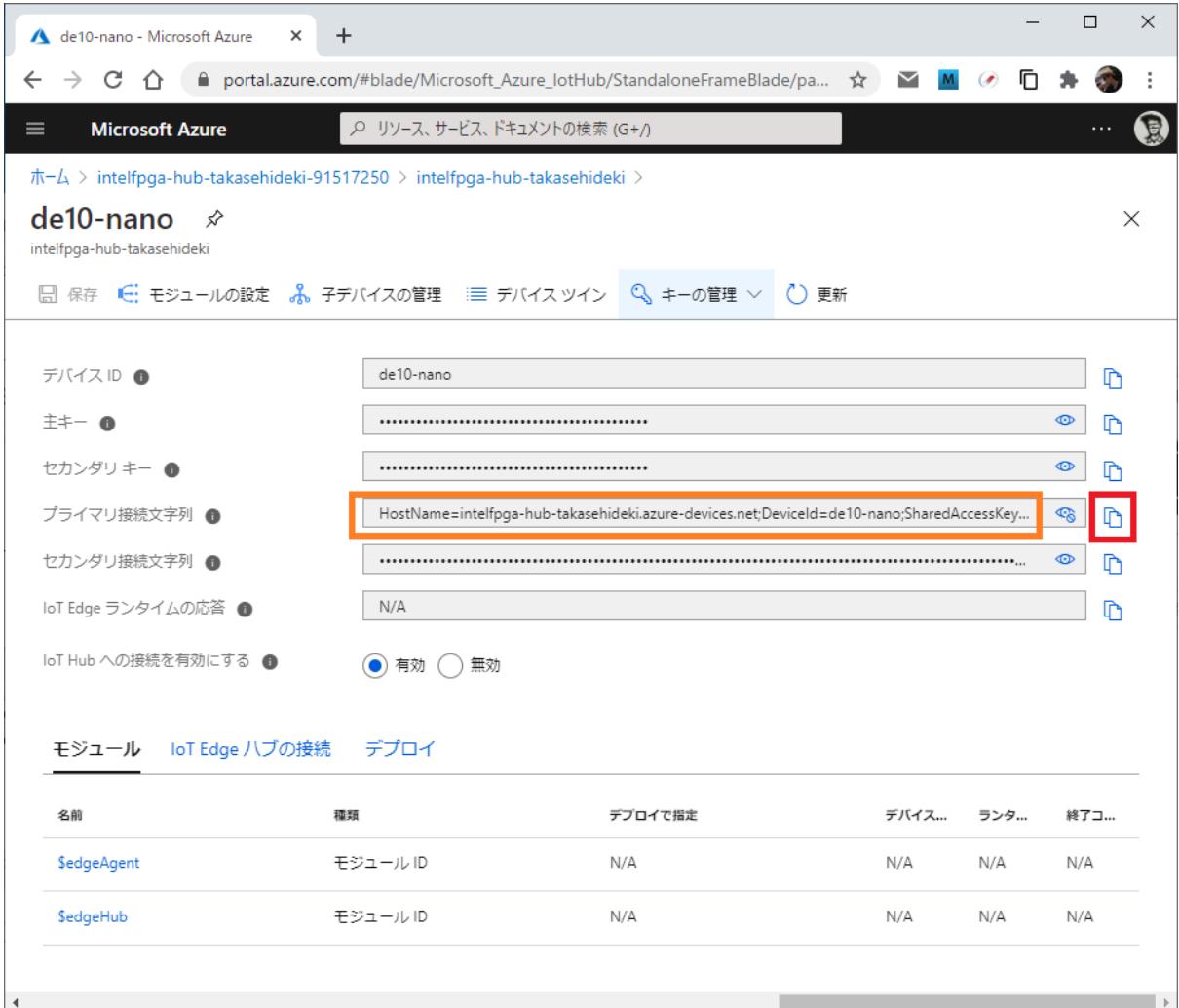


The screenshot shows the Microsoft Azure portal interface for managing IoT Edge devices. The left sidebar is titled 'IoT Hub' and includes sections for '組み込みのエンドポイント', 'フェールオーバー', 'プロパティ', 'ロック', 'テンプレートのエクスポート', 'クエリ エクスプローラー', 'IoT デバイス', and 'デバイスの自動管理' (with 'IoT Edge' selected). The main content area is titled 'intelfpga-hub-takasehideki | IoT Edge' and shows an informational message about IoT Edge devices. Below this, the 'IoT Edge デバイス' section is displayed, featuring a table for managing devices. The table has columns for 'デバイス ID' (Device ID), 'ランタイムの応答' (Runtime Response), 'IoT Edge モジュールの...' (IoT Edge Module of...), '接続済みクライアント...' (Connected Client...), and '展開' (Deployment). A row for the device 'de10-nano' is shown, with the 'デバイス ID' column highlighted by a red box.

デバイス ID	ランタイムの応答	IoT Edge モジュールの...	接続済みクライアント...	展開
de10-nano	該当なし	0	0	0

追加されたIoT Edgeデバイスの情報を確認します。

【プライマリ接続文字列】の情報が特に重要です。次に使用しますので、赤枠のアイコンをクリックしてエディタなどにコピーしておいてください。



The screenshot shows the Microsoft Azure IoT Hub Device blade for a device named 'de10-nano'. The blade is titled 'de10-nano' and shows the following fields:

デバイス ID	de10-nano	...
主キー
セカンダリ キー
プライマリ接続文字列	HostName=intelfpga-hub-takasehideki.azure-devices.net;DeviceId=de10-nano;SharedAccessKey=...	...
セカンダリ接続文字列
IoT Edge ランタイムの応答	N/A	...
IoT Hub への接続を有効にする	<input checked="" type="radio"/> 有効 <input type="radio"/> 無効	

Below the blade, there is a navigation bar with tabs: モジュール (selected), IoT Edge ハブの接続, and デプロイ.

Under the tabs, there is a table showing module information:

名前	種類	デプロイで指定	デバイス...	ランタ...	終了コ...
\$edgeAgent	モジュール ID	N/A	N/A	N/A	N/A
\$edgeHub	モジュール ID	N/A	N/A	N/A	N/A

ここまでのお作業に関するMicrosoftの公式ドキュメントは、下記をご参照ください。

参考: Azure Portal を使用して IoT Hub を作成する | Microsoft Docs

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

5. DE10-NanoボードのIoT Edgeデバイスとしての登録

次に、DE10-NanoボードをIoT Edgeデバイスとして登録します。

DE10-Nanoボードにシリアル／ssh接続して、ターミナルを立ち上げます。

まずはDE10-Nanoボードで稼働しているIoT Edge Runtimeの設定を行います。

なお、以降は本資料作成時点のAzure IoT Edge 1.0.9.4の情報となります。最新版では設定方法やログが異なることがあります。

IoT Edge Runtimeの設定情報は、DE10-Nanoボード上にある `/etc/iotedge/config.yaml` というファイルの編集によって行います。

ターミナル上でのファイルの編集には、Vimが使用できます。

Vimやターミナルの操作が苦手な方は、[準備篇F. 4.2 ssh接続](#) で紹介したようにTera Termでssh接続してから、【ファイル(F)】>【SSH SCP...】の機能を使うとよいでしょう。ホストのPCにDE10-Nanoボードの `/etc/iotedge/config.yaml` のファイルを[Receive]してから、メモ帳などのWindowsアプリで編集し、DE10-Nanoボードの `/etc/iotedge/` に[Send]してください。あるいは、[準備篇F. 4.3 リモートデスクトップ接続](#)で紹介したように、リモートデスクトップを開いてLeafpadを使うこともできます。

編集すべき箇所は55行目の `"<ADD DEVICE CONNECTION STRING HERE>"` のところです。

```
(省略)
# Manual provisioning configuration
provisioning:
  source: "manual"
  device_connection_string: "<ADD DEVICE CONNECTION STRING HERE>"

# DPS TPM provisioning configuration
# provisioning:
(省略)
```

例えば【プライマリ接続文字列】が下記でしたら、

`HostName=intelfpga-hub-takasehideki.azure-devices.net;DeviceId=de10nano;SharedAccessKey=ABCdefg1234HIJklmn5678PQrstu`

55行目を下記のように修正します。文字列の先頭と末尾に `"` が必要なことに注意してください。

```
device_connection_string: "HostName=intelfpga-hub-takasehideki.azure-devices.net;DeviceId=de10nano;SharedAccessKey=ABCdefg1234HIJklmn5678PQrstu"
```

下記には、Vimでの操作を簡単に示します。

```
# vimでファイルを開く
root@de10nano:~# vim /etc/iotedge/config.yaml
### '/' で "device_connection_string" を検索するか、'55+G' で55行目に移動する
### このファイルの権限は 600 なので、保存時には ':w!' で強制書き込みが必要
```

```
# 記入例の確認
root@de10nano:~# grep device_connection_string /etc/iotedge/config.yaml
device_connection_string: "HostName=intelfpga-hub-takasehideki.azure-devices.net;DeviceId=de10nano;SharedAccessKey=ABCdefg1234HIJKLMNOPQRSTU"
```

次に、コンテナ向けのDNSを設定します。

`/etc/docker/daemon.json` のファイルを、新規作成して、下記を記述してください。

(Tera TermでSSH SCPされる方は、このファイルは最初は存在していませんので、ホストPCで作成したファイルをDE10-Nanoボードの `/etc/docker/` に[Send]してください)

```
{
  "dns": ["8.8.8.8", "8.8.4.4"],
  "log-driver": "json-file",
  "log-opt": {
    "max-size": "10m",
    "max-file": "3"
  }
}
```

IoT Edgeのサービスを再起動して、その稼働状況を確認します。

下記の通り `active (running)` という表示が出ていればOKです。

```
# iotedgeサービスを再起動する
root@de10nano:~# systemctl restart iotedge
# iotedgeサービスの稼働状況を確認する
root@de10nano:~# systemctl status iotedge
● iotedge.service - Azure IoT Edge daemon
  Loaded: loaded (/lib/systemd/system/iotedge.service; enabled; vendor preset:
  Active: active (running) since Tue 2020-09-15 09:23:17 UTC; 3s ago
    Docs: man:iotedged(8)
  Main PID: 2312 (iotedged)
    Tasks: 2 (limit: 2325)
   CGroup: /system.slice/iotedge.service
           mq2312 /usr/bin/iotedged -c /etc/iotedge/config.yaml

Sep 15 09:23:17 de10nano systemd[1]: Started Azure IoT Edge daemon.
Sep 15 09:23:17 de10nano iot edged[2312]: 2020-09-15T09:23:17Z [INFO] - Starting
Sep 15 09:23:17 de10nano iot edged[2312]: 2020-09-15T09:23:17Z [INFO] - Version -
Sep 15 09:23:17 de10nano iot edged[2312]: 2020-09-15T09:23:17Z [INFO] - Using con
Sep 15 09:23:17 de10nano iot edged[2312]: 2020-09-15T09:23:17Z [INFO] - Configuri
Sep 15 09:23:17 de10nano iot edged[2312]: 2020-09-15T09:23:17Z [INFO] - Configuri
Sep 15 09:23:17 de10nano iot edged[2312]: 2020-09-15T09:23:17Z [INFO] - Transpare
Sep 15 09:23:17 de10nano iot edged[2312]: 2020-09-15T09:23:17Z [INFO] - Finished
Sep 15 09:23:17 de10nano iot edged[2312]: 2020-09-15T09:23:17Z [INFO] - Initializ
lines 1-18/18 (END)
# 'q'で確認を終了する
```

※トラブルシューティング:

ここで、`Active: failed (Result: exit-code)` となることがあります。原因としては、`/etc/iotedge/config.yaml` の編集を行っていないことが考えられます。少し戻って編集してください。その後、`systemctl restart iotedge` を実行して IoT Edge のサービスを再起動してください。

IoT Hubとの疎通を確認してみましょう。

初回の IoT Edge Runtime 起動時には、システムモジュールの取得や展開などでちょっと時間が掛かります。ちょっと待ってから、`iotedge check` というコマンドを実行します。

3個の警告と1個のエラーメッセージ(edgeHubに関するもの)が出ますが,,, ここではまだ無視して問題ありません。

(エラーは IoT Edge デバイスのデプロイ構成がまだなにも設定されていないことに起因しています)

```
# IoT Hubとの疎通を確認する
root@de10nano:~# iotedge check
Configuration checks
-----
✓ config.yaml is well-formed - OK
✓ config.yaml has well-formed connection string - OK
✓ container engine is installed and functional - OK
✓ config.yaml has correct hostname - OK
✓ config.yaml has correct URIs for daemon mgmt endpoint - OK
? latest security daemon - Warning
    Installed IoT Edge daemon has version 1.0.9.4 but 1.0.9.5 is the latest
    stable version available.
    Please see https://aka.ms/iotedge-update-runtime for update instructions.
✓ host time is close to real time - OK
✓ container time is close to host time - OK
✓ DNS server - OK
? production readiness: certificates - Warning
    The Edge device is using self-signed automatically-generated development
    certificates.
    They will expire in 89 days (at 2020-12-14 09:23:30 UTC) causing
    module-to-module and downstream device communication to fail on an active
    deployment.
    After the certs have expired, restarting the IoT Edge daemon will trigger it
    to generate new development certs.
    Please consider using production certificates instead. See
    https://aka.ms/iotedge-prod-checklist-certs for best practices.
✓ production readiness: container engine - OK
✓ production readiness: logs policy - OK
? production readiness: Edge Agent's storage directory is persisted on the host
    filesystem - Warning
    The edgeAgent module is not configured to persist its /tmp/edgeAgent
    directory on the host filesystem.
    Data might be lost if the module is deleted or updated.
    Please see https://aka.ms/iotedge-storage-host for best practices.
✗ production readiness: Edge Hub's storage directory is persisted on the host
```

```

filesystem - Error
    Could not check current state of edgeHub container

Connectivity checks
-----
✓ host can connect to and perform TLS handshake with IoT Hub AMQP port - OK
✓ host can connect to and perform TLS handshake with IoT Hub HTTPS / WebSockets
port - OK
✓ host can connect to and perform TLS handshake with IoT Hub MQTT port - OK
✓ container on the default network can connect to IoT Hub AMQP port - OK
✓ container on the default network can connect to IoT Hub HTTPS / WebSockets
port - OK
✓ container on the default network can connect to IoT Hub MQTT port - OK
✓ container on the IoT Edge module network can connect to IoT Hub AMQP port - OK
✓ container on the IoT Edge module network can connect to IoT Hub HTTPS /
WebSockets port - OK
✓ container on the IoT Edge module network can connect to IoT Hub MQTT port - OK

19 check(s) succeeded.
3 check(s) raised warnings. Re-run with --verbose for more details.
1 check(s) raised errors. Re-run with --verbose for more details.

```

※トラブルシューティング:

ここで、上記以外で**2個以上のエラー**または**4個以上の警告**のメッセージが表示される場合は、下記の原因が考えられます。

① `/etc/iotedge/config.yaml` において編集した55行目

```
device_connection_string: "<ADD DEVICE CONNECTION STRING HERE>"
```

に「プライマリ接続文字列」が正しく貼り付けされていないことが考えれます。少し戻って編集内容を見直してみてください。その後、`systemctl restart iotedge` を実行してIoT Edgeのサービスを再起動してください。

② コンテナ向けのDNS設定 `/etc/docker/daemon.json` が正しく作成・編集できていないことが考えられます。少し戻って編集内容を見直してみてください。その後、

`systemctl restart iotedge` を実行してIoT Edgeのサービスを再起動してください。

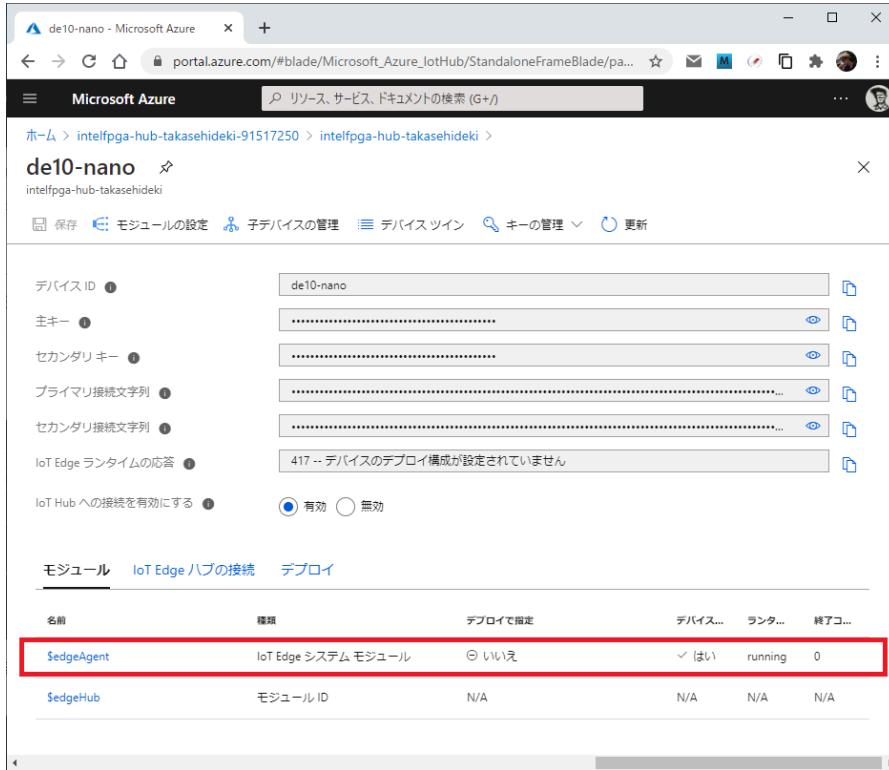
IoT Edge Runtimeのシステムモジュールの1つである `edgeAgent` が配置されました。

```

# IoT Edgeモジュールの動作状況を確認する
root@de10nano:~# iotedge list
NAME          STATUS        DESCRIPTION      CONFIG
edgeAgent     running      Up 24 minutes
mcr.microsoft.com/azureiotedge-agent:1.0

```

Azure PortalからもIoT Edgeデバイスの動作状況を確認してみましょう。
(このページは、IoT Hubリソースページの左横メニューから「デバイスの自動管理」> [IoT Edge]を選択して、"de10-nano" をクリックして開きます)



The screenshot shows the Azure Portal interface for managing an IoT Edge device. The device ID is 'de10-nano'. The 'Modules' section is highlighted with a red box. The table shows the following data:

名前	種類	デプロイで指定	デバイス...	ランタ...	終了コ...
\$edgeAgent	IoT Edge システム モジュール	⊕ いいえ	✓ (はい)	running	0
\$edgeHub	モジュール ID	N/A	N/A	N/A	N/A

6. Option: Azure PortalからのIoT Edgeモジュールのデプロイ

IoT Edgeデバイスとして登録されたDE10-Nanoボードに、IoT Edgeモジュールをデプロイしてみましょう。

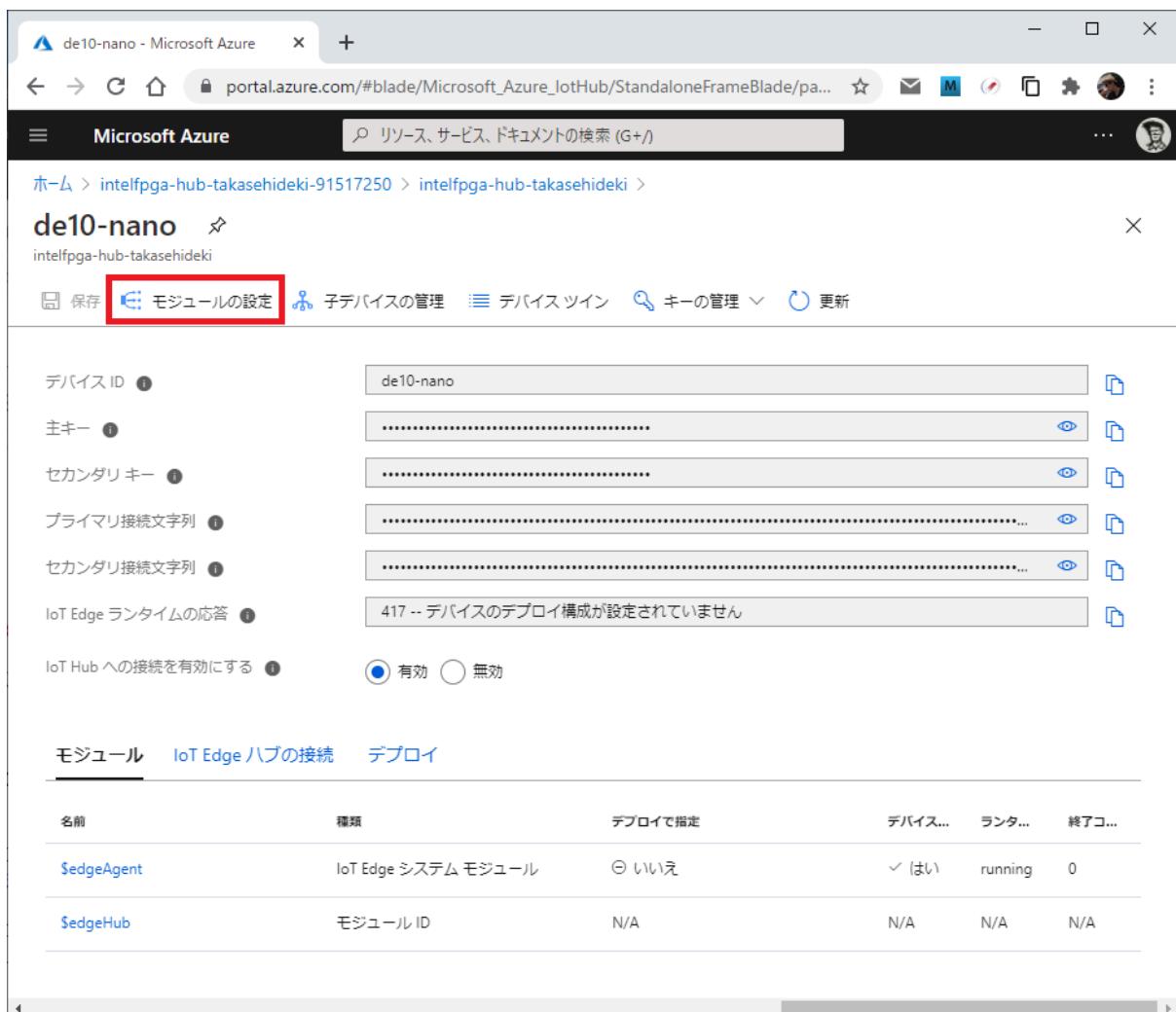
今回は、Docker Hubで講師が公開している出来合いのコンテナイメージを使用して、これをIoT Edgeモジュールとして実行します。このモジュールは、1秒ごとにG-Sensorで加速度の3次元の値を取得して、この値をIoT Hubにメッセージとして300回送信します。

takasehideki/de10nano-g-sensor - Docker Hub
<https://hub.docker.com/r/takasehideki/de10nano-g-sensor>

IoT Edgeデバイスにモジュールをデプロイする方法は、主に、Azure Portalからの操作と、VS Codeでの操作の2種類があります。ここでは、前者の方法を紹介します。（後者の方法は、[E.7 . Option: VS CodeからのIoT Edgeモジュールのデプロイ](#)で紹介します）

Azure Portalにて、IoT Hubリソースページの左横メニューから「デバイスの自動管理」> [IoT Edge]を選択して、"de10-nano" をクリックしてください。

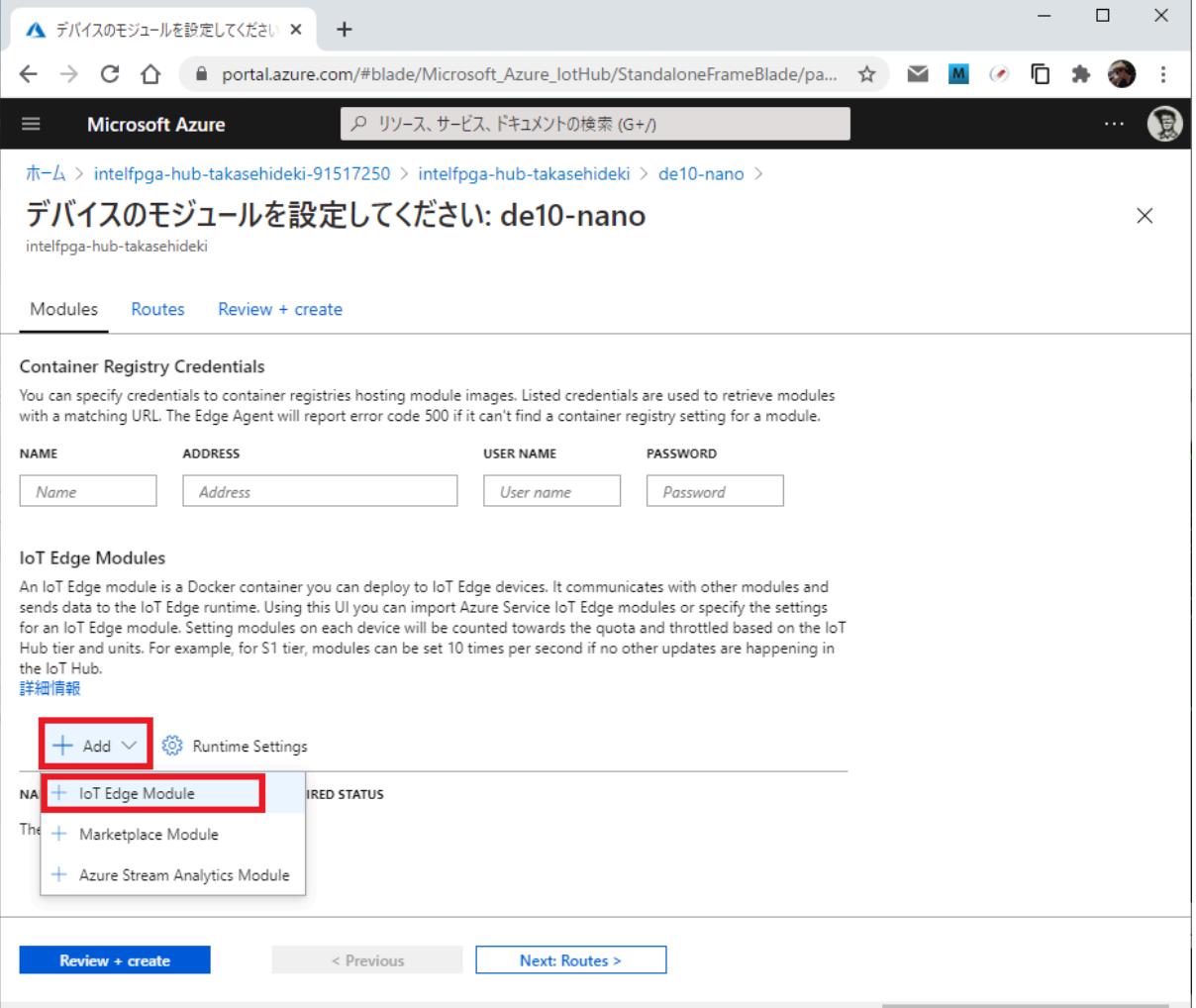
【モジュールの設定】をクリックして開きます。



名前	種類	デプロイで指定	デバイス...	ランタ...	終了コ...
\$edgeAgent	IoT Edge システム モジュール	⊕ いいえ	✓ はい	running	0
\$edgeHub	モジュール ID	N/A	N/A	N/A	N/A

[デバイスのモジュールを設定してください: de10-nano]の画面になります.

- **+ Add** をクリックし、続けて **+ IoT Edge Module** をクリックします。



The screenshot shows the Microsoft Azure IoT Hub Standalone Frame blade. The URL in the browser is https://portal.azure.com/#blade/Microsoft_Azure_IotHub/StandaloneFrameBlade/pa.... The page title is "デバイスのモジュールを設定してください: de10-nano". The main navigation tabs are "Modules", "Routes", and "Review + create".

Container Registry Credentials
You can specify credentials to container registries hosting module images. Listed credentials are used to retrieve modules with a matching URL. The Edge Agent will report error code 500 if it can't find a container registry setting for a module.

NAME	ADDRESS	USER NAME	PASSWORD
<input type="text" value="Name"/>	<input type="text" value="Address"/>	<input type="text" value="User name"/>	<input type="text" value="Password"/>

IoT Edge Modules
An IoT Edge module is a Docker container you can deploy to IoT Edge devices. It communicates with other modules and sends data to the IoT Edge runtime. Using this UI you can import Azure Service IoT Edge modules or specify the settings for an IoT Edge module. Setting modules on each device will be counted towards the quota and throttled based on the IoT Hub tier and units. For example, for S1 tier, modules can be set 10 times per second if no other updates are happening in the IoT Hub.

[詳細情報](#)

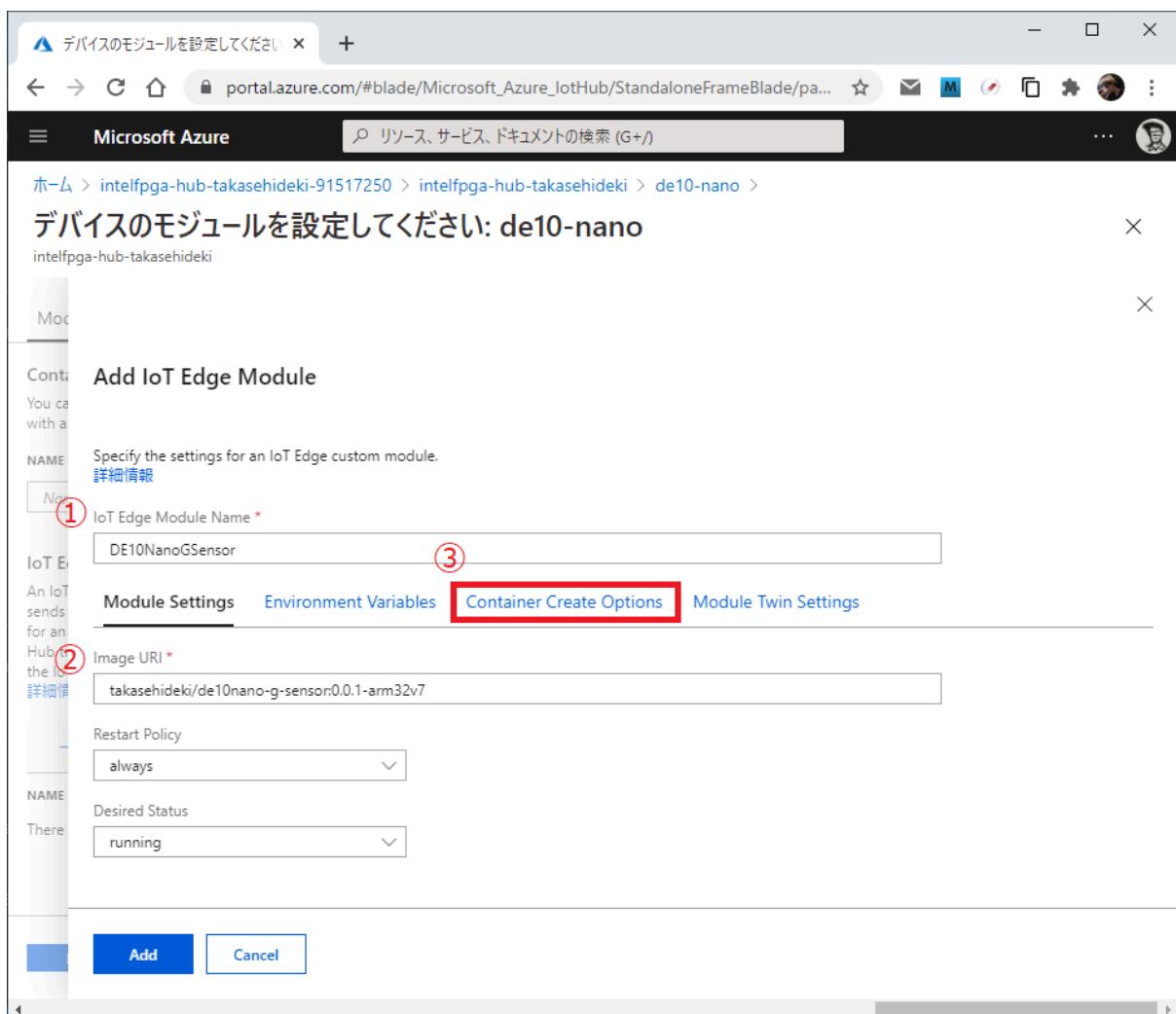
Add **Runtime Settings**

NAME	STATUS
+ IoT Edge Module	READY
+ Marketplace Module	READY
+ Azure Stream Analytics Module	READY

[Review + create](#) [Next: Routes >](#)

「Add IoT Edge Module」画面になります。

- ①【IoT Edge Module Name】は、 **DE10NanoGSensor** を入力してください。
- 【Module Settings】タブ
 - ②【Image URI】は、 **takasehideki/de10nano-g-sensor:0.0.1-arm32v7** を入力します。
 - **docker pull** においてイメージのURIを指定することに相当します。
 - 【Restart Policy】では、モジュールの障害時の動作を指定します。
always (常に再起動する)のままとしてください。
 - 【Desired Status】は、デプロイ時の状態を指定します。
running (デプロイ時に実行する)のままとしてください。
- 次は【Container Create Options】の設定を行います。
③のところをクリックして移動してください。



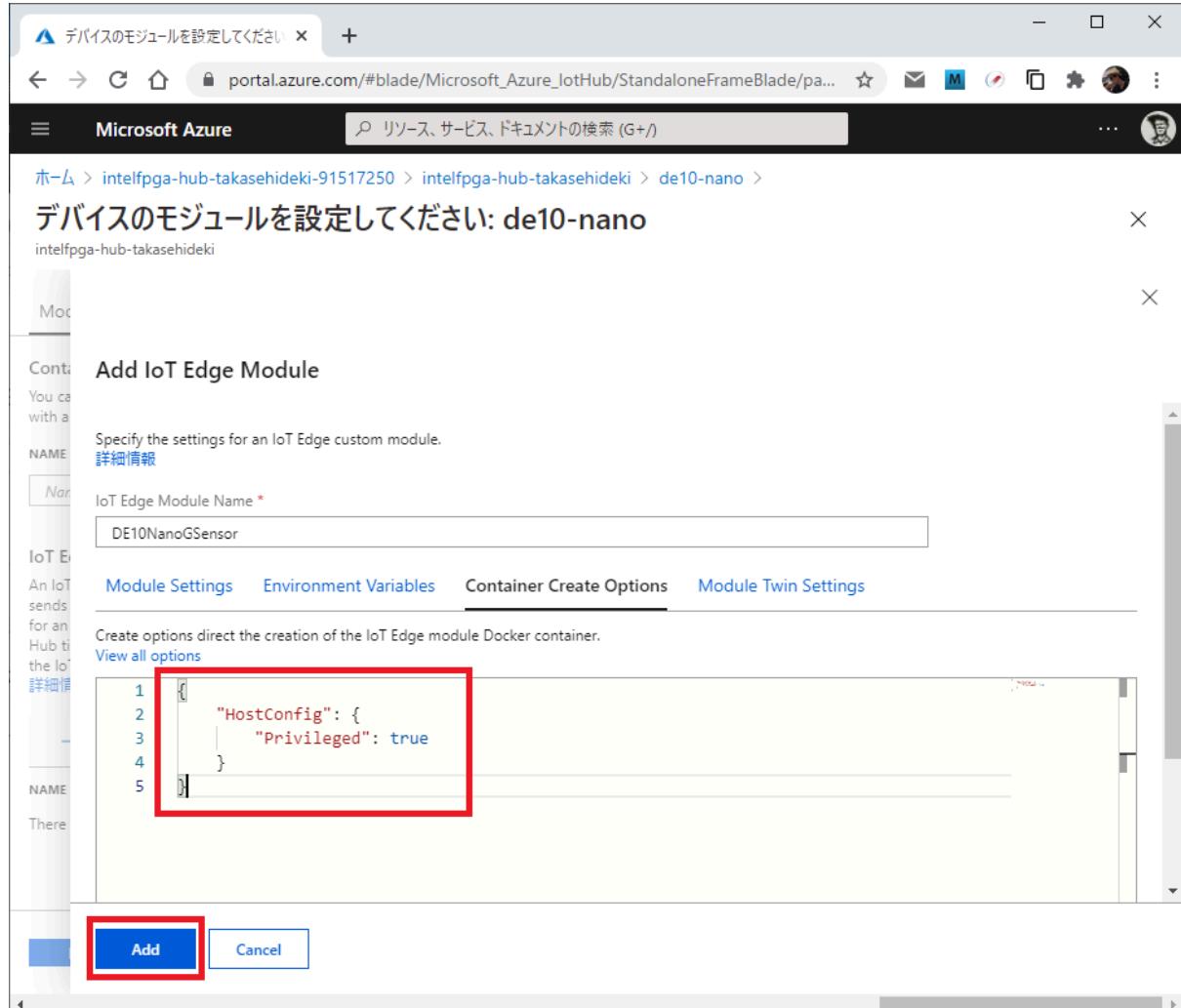
「Add IoT Edge Module」画面の続きです。

- [Container Create Options] タブ
 - フィールドに次の記述を入力してください。

```
{  
  "HostConfig": {  
    "Privileged": true  
  }  
}
```

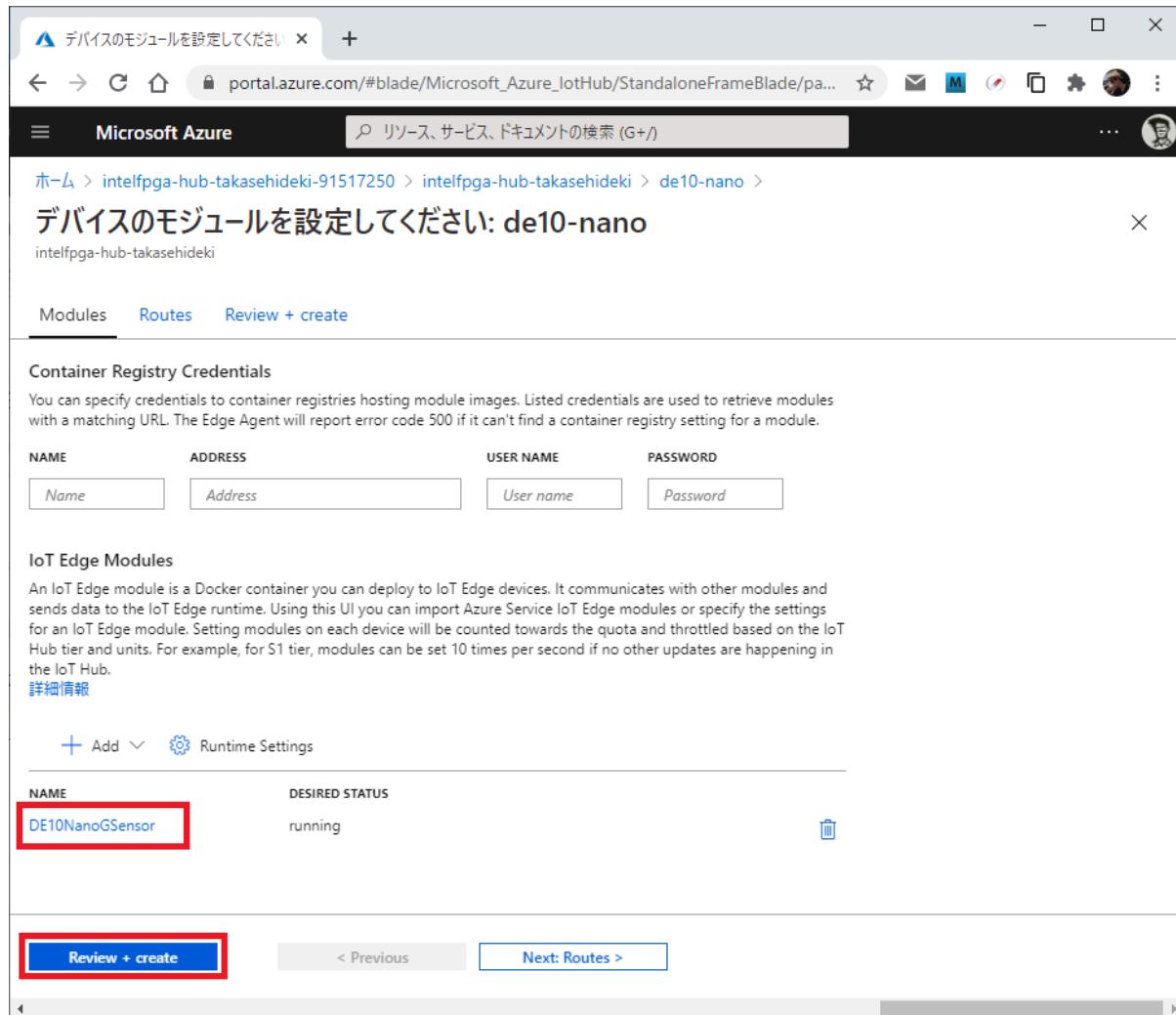
- コンテナイメージの作成時のオプションをJSON形式で指定しています。
- 今回のモジュールはG-Sensorの接続されている `/dev/i2c-1` のデバイスにアクセスします。そのアクセス権限(実行特権)を与えています。
- Dockerコマンドのオプション `--privileged` に相当します。

- 下部の **Add** をクリックします。



最初の画面に戻ります。

- IoT Edge Modules の一覧に **DE10NanoGSensor** が表示されていることを確認してください。
- 下部の **【Review + create】** をクリックします。



デバイスのモジュールを設定してください: de10-nano

Container Registry Credentials

IoT Edge Modules

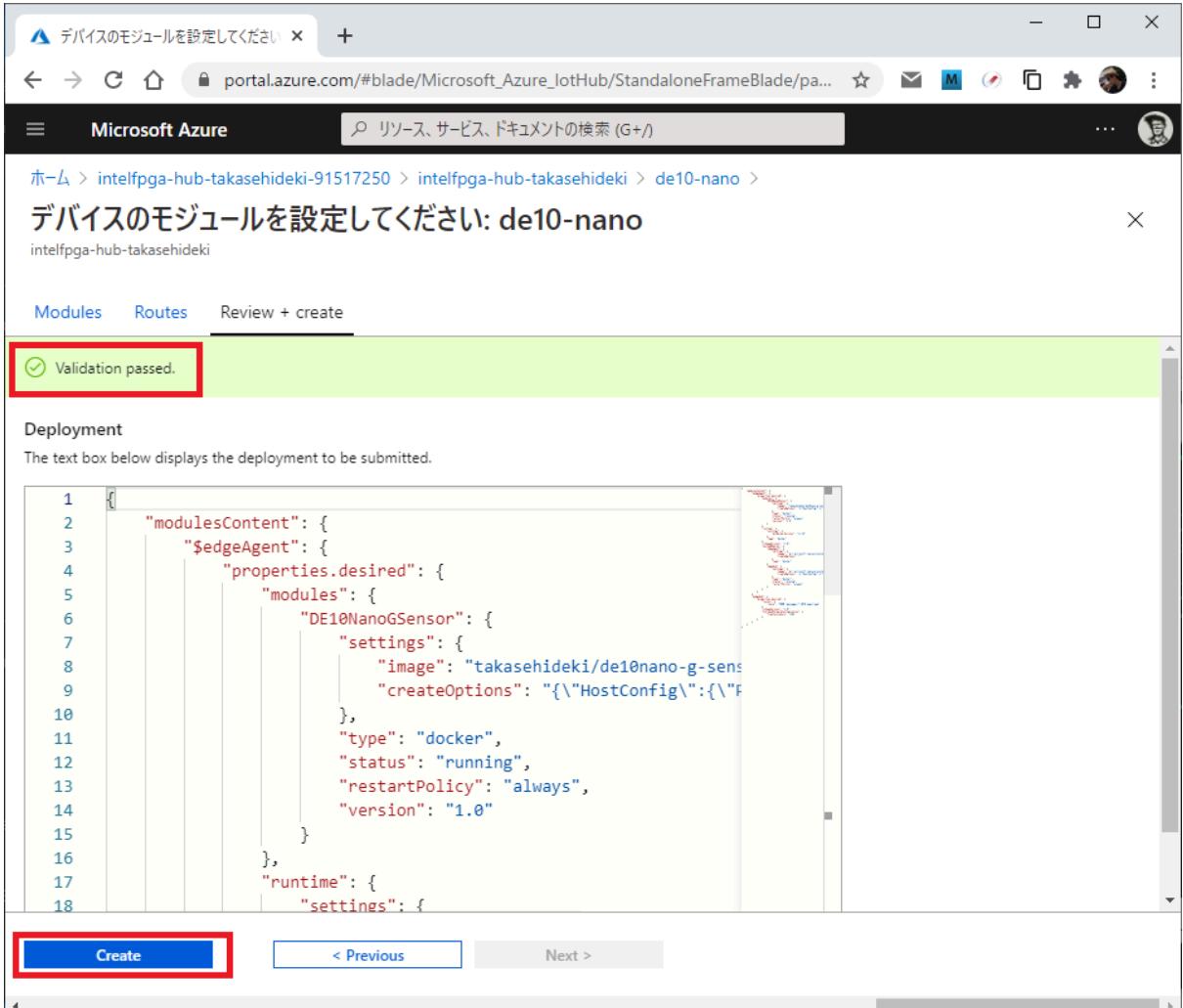
NAME	ADDRESS	USER NAME	PASSWORD
<input type="text" value="Name"/>	<input type="text" value="Address"/>	<input type="text" value="User name"/>	<input type="text" value="Password"/>

DE10NanoGSensor running

Review + create

デプロイの設定の最終確認をします。

- "Validation passed." と緑色のメッセージが出ていることを確認してください。
- 下部の **[Create]** をクリックします。



デバイスのモジュールを設定してください: de10-nano

Validation passed.

Deployment

The text box below displays the deployment to be submitted.

```
1  {
2      "modulesContent": {
3          "$edgeAgent": {
4              "properties.desired": {
5                  "modules": {
6                      "DE10NanoGSensor": {
7                          "settings": {
8                              "image": "takasehideki/de10nano-g-sensor:1.0",
9                              "createOptions": "{\"HostConfig\":{\"PortBindings\":[{\"HostPort\": 80, \"ContainerPort\": 80}],\"Binds\":[\"/dev/bus/usb:/dev/bus/usb\"]}}",
10                         },
11                         "type": "docker",
12                         "status": "running",
13                         "restartPolicy": "always",
14                         "version": "1.0"
15                     },
16                     "runtime": {
17                         "settings": {
18                         }
19                     }
20                 }
21             }
22         }
23     }
24 }
```

Create

Dockerコンテナイメージが展開され、DE10-Nanoボードで稼働します。
 しばらく待ってから下記の画面の右上にある【更新】をクリックしてください。
 すると、下部の【モジュール】の一覧に **DE10NanoGSensor** が表示されます。そして、その【ランタイムの状態】が "running" になっていることが確認できます。

名前	種類	デプロイで指定	デバイ...	ランタ...	終了コ...
\$edgeAgent	IoT Edge システム モジュール	✓ はい	✓ はい	running	0
\$edgeHub	IoT Edge システム モジュール	✓ はい	✓ はい	running	0
DE10NanoGSensor	IoT Edge のカスタム モジュール	✓ はい	✓ はい	running	0

DE10-Nanoボード上のターミナルで動作を確認してみましょう。
 IoT Edge Runtimeのシステムモジュールである `edgeHub` が配置されており、さらに今回デプロイした **DE10NanoGSensor** も展開されたことが表示されます。

```
# IoT Edgeモジュールの一覧と動作状況を確認する
root@de10nano:~# iotedge list
NAME          STATUS        DESCRIPTION      CONFIG
DE10NanoGSensor  running      Up 4 minutes
takasehideki/de10nano-g-sensor:0.0.1-arm32v7
edgeAgent      running      Up 2 hours
mcr.microsoft.com/azureiotedge-agent:1.0
edgeHub        running      Up 44 minutes
mcr.microsoft.com/azureiotedge-hub:1.0
```

このモジュールの動作を再掲します。1秒ごとにG-Sensorで加速度の3次元の値を取得します。そして、その値をIoT Hubにメッセージとして300回送信します。

`iotedge logs <モジュール名>` コマンドで、対象モジュールの実行ログ(`printf`メッセージ)を確認できます。`DE10NanoGSensor` のログを表示してみましょう。危険が危なくない範囲でボードを振ってみると楽しいでしょう。

```
# IoT Edgeモジュールの動作ログを確認する
root@de10nano:~# iotedge logs DE10NanoGSensor | tail
IoTHubModuleClient_LL_SendEventAsync accepted message [15] for transmission to
IoT Hub.
Confirmation[14] received for message tracking id = 14 with result =
IOTHUB_CLIENT_CONFIRMATION_OK
[17]X=-20 mg, Y=-12 mg, Z=1004 mg
IoTHubModuleClient_LL_SendEventAsync accepted message [16] for transmission to
IoT Hub.
Confirmation[15] received for message tracking id = 15 with result =
IOTHUB_CLIENT_CONFIRMATION_OK
[18]X=-20 mg, Y=-12 mg, Z=1000 mg
IoTHubModuleClient_LL_SendEventAsync accepted message [17] for transmission to
IoT Hub.
Confirmation[16] received for message tracking id = 16 with result =
IOTHUB_CLIENT_CONFIRMATION_OK
[19]X=-16 mg, Y=-12 mg, Z=1004 mg
IoTHubModuleClient_LL_SendEventAsync accepted message [18] for transmission to
IoT Hub.
```

※トラブルシューティング：

このとき、下記のようにFailedしている場合は、「**Add IoT Edge Module**」画面での[**Container Create Options**]タブの記述にtypoがあった可能性が高いです。Portal画面で見直してみて、デプロイを再度行ってみてください。

```
root@de10nano:~# iotedge logs DE10NanoGSensor | tail
(省略)
Failed to open the i2c bus of gsensor: No such file or directory
===== gsensor test =====
Failed to open the i2c bus of gsensor: No such file or directory
===== gsensor test =====
```

なお、IoT Edgeモジュールはコンテナとして実行されています。IoT Edgeでは、コンテナエンジンとしてOSSである [moby-engine](#) が採用されています。

参考：サポートされているオペレーティング システム、コンテナー エンジン - Azure IoT Edge | Microsoft Docs

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

つまり、Dockerの一般的なコマンドで、コンテナやイメージの状態を確認することもできます。

```
# イメージの一覧を表示する
root@de10nano:~# docker images
REPOSITORY          TAG      IMAGE ID
CREATED            SIZE
takasehideki/de10nano-g-sensor    0.0.1-arm32v7      0a1edabe8550
3 days ago         187MB
mcr.microsoft.com/azureiotedge-hub 1.0      d6d6a399d448
5 days ago         224MB
mcr.microsoft.com/azureiotedge-agent 1.0      fb8e08219594
5 days ago         210MB
mcr.microsoft.com/azureiotedge-diagnostics 1.0.9.4    1acd9bbac7da
8 weeks ago        7.25MB

# コンテナの一覧を表示する('docker container ls' と同様)
root@de10nano:~# docker ps
CONTAINER ID        IMAGE               COMMAND
CREATED            STATUS              PORTS
NAMES
94a7d2a2e4bc      takasehideki/de10nano-g-sensor:0.0.1-arm32v7
"../iothub_client_sam..."  10 minutes ago   Up 10 minutes
DE10NanoGSensor
6d0351d98594      mcr.microsoft.com/azureiotedge-hub:1.0      "/bin/sh -c
'echo \"$...\"  51 minutes ago   Up 51 minutes    0.0.0.0:443->443/tcp,
0.0.0.0:5671->5671/tcp, 0.0.0.0:8883->8883/tcp  edgeHub
79055a382f6c      mcr.microsoft.com/azureiotedge-agent:1.0      "/bin/sh -c
'echo \"$...\"  2 hours ago     Up 2 hours
edgeAgent

# Dockerコンテナの資源使用量を確認する('--no-stream' は一度のみ表示)
root@de10nano:~# docker stats --no-stream
CONTAINER ID        NAME            CPU %      MEM USAGE / LIMIT
MEM %              NET I/O        BLOCK I/O  PIDS
94a7d2a2e4bc      DE10NanoGSensor 0.01%      1.285MiB / 1002MiB
0.13%              36.7kB / 97.1kB 0B / 0B    1
6d0351d98594      edgeHub        0.13%      82.01MiB / 1002MiB
8.19%              339kB / 297kB 0B / 0B    24
79055a382f6c      edgeAgent      0.00%      24.31MiB / 1002MiB
2.43%              60.5kB / 47.4kB 0B / 0B    17
```

ちなみに、IoT Hubとの疎通を **iotedge check** で再度確認してみましょう。

IoT Edgeデバイスのデプロイ構成が設定されてモジュールが配置されましたので、先ほどの edgeHubに関するエラーメッセージは無くなりました(警告に変わりました)

```
# IoT Hubとの疎通を確認する
root@de10nano:~# iotedge check
Configuration checks
-----
(省略)
✓ production readiness: logs policy - OK
```

```
? production readiness: Edge Agent's storage directory is persisted on the host
filesystem - Warning
    The edgeAgent module is not configured to persist its /tmp/edgeAgent
    directory on the host filesystem.
        Data might be lost if the module is deleted or updated.
        Please see https://aka.ms/iotedge-storage-host for best practices.
? production readiness: Edge Hub's storage directory is persisted on the host
filesystem - Warning
    The edgeHub module is not configured to persist its /tmp/edgeHub directory
    on the host filesystem.
        Data might be lost if the module is deleted or updated.
        Please see https://aka.ms/iotedge-storage-host for best practices.

Connectivity checks
-----
(省略)
19 check(s) succeeded.
4 check(s) raised warnings. Re-run with --verbose for more details.
```

7. Tips: DE10NanoGSensorのソースコード

出来合いの品として今回利用したモジュールのソースコードは、下記で公開しています。時間があれば解説したいと思います。

algyan/IntelFPGA_AzureOTA at vsc_g-sensor
https://github.com/algyan/IntelFPGA_AzureOTA/tree/vsc_g-sensor

ベースに下記を使用して実装しました。[commit log](#)を追ってみたらなにをやったのか分かって楽しいかもしれません。

- SystemCD:
DE10-Nano_v.1.3.8_HWrevC_SystemCD\Demonstrations\SoC\hps_gsensord
- azure-iot-sdk-c\iothub_client\samples\iothub_client_sample_module_sender at
master · Azure/azure-iot-sdk-c
https://github.com/Azure/azure-iot-sdk-c/tree/master/iothub_client/samples/iothub_client_sample_module_sender

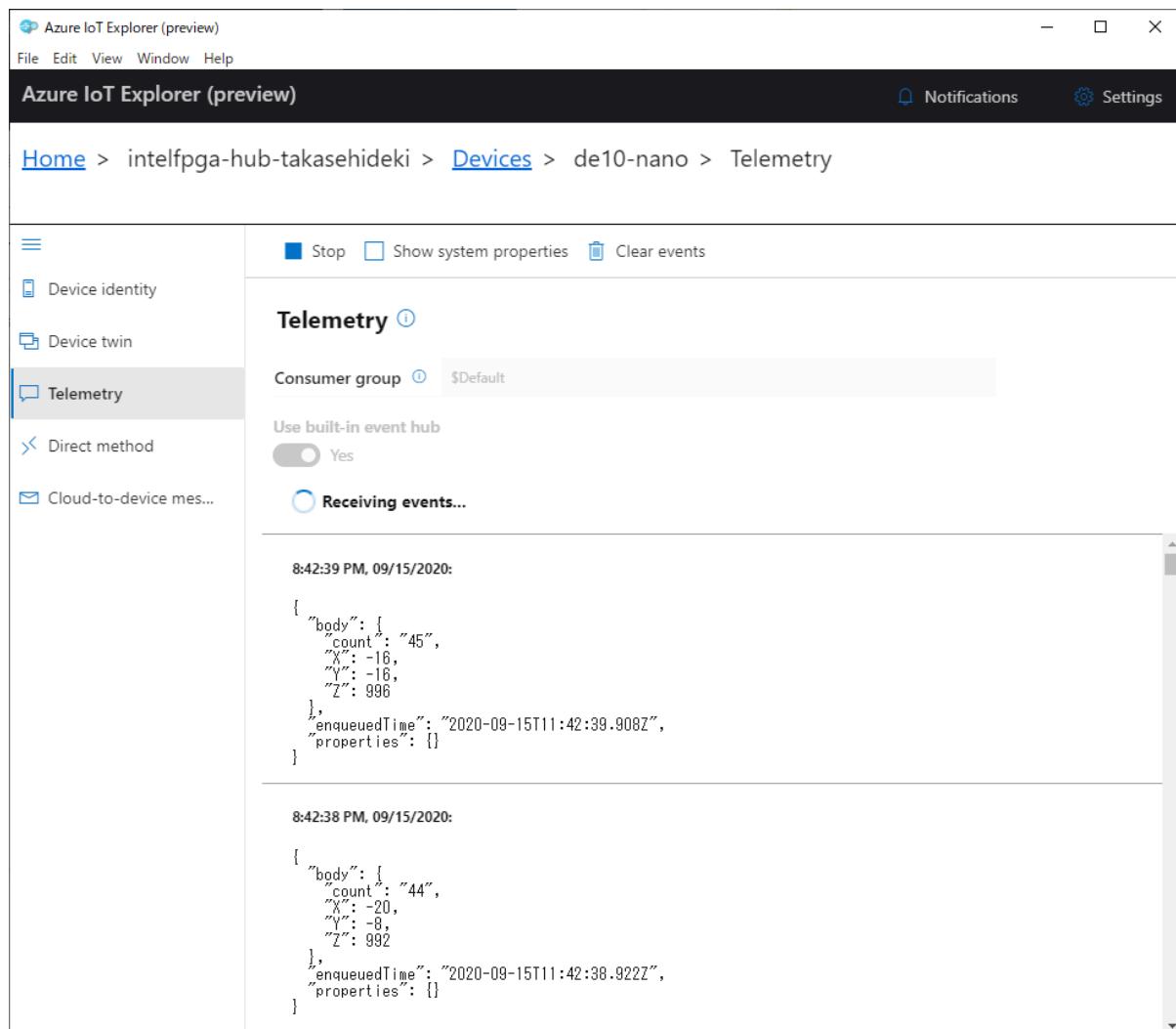
8. Tips: Azure IoT エクスプローラーでIoT Hubの受信メッセージを確認する

IoT Edgeデバイスから送信したメッセージ(IoT Hubの受信したメッセージ)を確認するには、Azure IoT Explorerを使用するのがおすすめです。

Azure IoT エクスプローラーをインストールして使用する | Microsoft Docs

<https://docs.microsoft.com/ja-jp/azure/iot-pnp/howto-use-iot-explorer>

IoT Hubの共有アクセスポリシー "iothubowner" の接続文字列を設定するだけで、すぐに使えます。まだpreview版ではありますが、IoT Plug and Playにすでに対応していますし、macOSでも動作するという嬉しさもあります。



次の章に進む:[D. Visual Studio Codeの設定](#)

目次に戻る:[A. 3. ハンズオンの内容](#)