

IoT Edge ワークショップ

Global Blackbelt Sales Microsoft Corporation

Sept-2018

Table of Contents

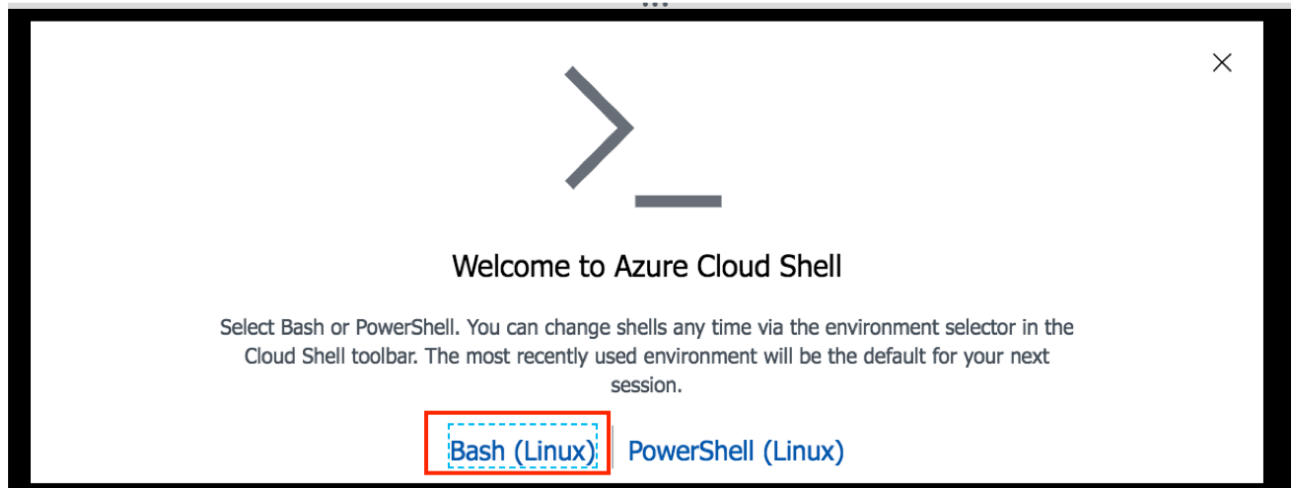
演習: Cloud Shell (Bash)の設定	1
タスク1: Cloud Shellの初期設定	1
演習: Azure IoT Edge デバイスの作成	2
タスク1 : Linux VMのデプロイ	2
タスク2 : コンテナランタイムのインストール	5
タスク3 : IoT Edgeデバイスの追加	7
タスク4 : IoT Edgeランタイムのインストール	8
タスク5 : IoT Edgeデバイスにシミュレーションモジュールを追加	10

演習: Cloud Shell (Bash)の設定

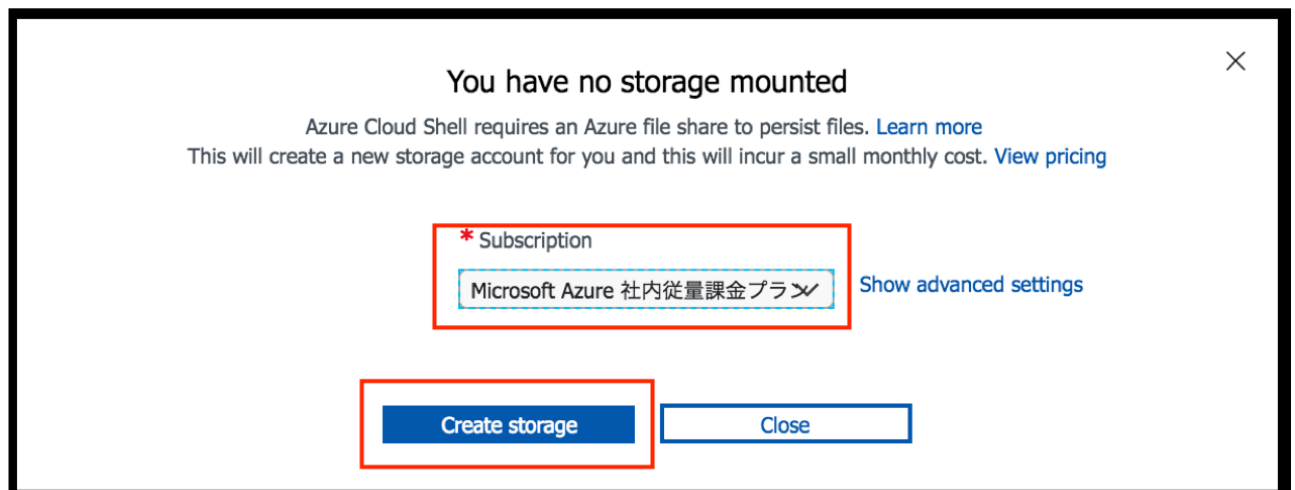
Cloud Shell (Bash) 環境を設定します。

タスク1: Cloud Shellの初期設定

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



3. 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。



4. 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。

```
Bash ▾ | 🔌 ? ⚙️ 📄 📋 {}  
Resource group: cloud-shell-storage-southeastasia  
Storage account: cs-██████████████████████b  
File share: cs-████████████████████████████████████████6  
  
Initializing your account for Cloud Shell...\nRequesting a Cloud Shell.Succeeded.  
Connecting terminal...  
  
Welcome to Azure Cloud Shell  
  
Type "az" to use Azure CLI 2.0  
Type "help" to learn about Cloud Shell  
  
kei@Azure:~$
```

5. 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。

```
Bash ▾ | 🔌 ? ⚙️ 📄 📋 {}  
Welcome to Azure Cloud Shell  
  
Type "az" to use Azure CLI 2.0  
Type "help" to learn about Cloud Shell  
  
kei@Azure:~$ ssh myadmin@[IP addr] ← コピーした接続文字列  
The authenticity of host '[IP addr] ([IP addr])' can't be established.  
ECDSA key fingerprint is SHA256:aaaaaaaa/bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb/0.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added '[IP addr]' (ECDSA) to the list of known hosts.  
myadmin@[IP addr]'s password: ← パスワードを入力 (表示はされない)  
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-1022-azure x86_64)  
  
* Documentation: https://help.ubuntu.com
```

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"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。

🔍 ubuntu ×

結果

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

1. [基本設定の構成]で各種項目を設定します。

高可用性

可用性ゾーン ⓘ

なし



選択した場所で利用できる Availability Zones はありません。Availability Zones をサポートしている場所を表示するには、aka.ms/zonedregions に移動します

* 可用性セット ⓘ



なし

Storage

マネージド ディスクを使用 ⓘ

いいえ

はい

OS ディスク サイズ ⓘ

既定のサイズ (30 GiB)



ネットワーク

* Virtual Network ⓘ



iothandson-vnet

* サブネット ⓘ



default (172.16.1.0/24)

* パブリック IP アドレス ⓘ



(新規) edgeDevice-ip

ネットワーク セキュリティ グループ ⓘ

Basic

詳細

* パブリック受信ポートを選択 ⓘ

0 項目が選択されました



+ 追加

OK

Table 1. 設定項目と設定値

設定項目	設定値
名前	任意のマシン名
VMディスクの種類	HDDまたはSSD
ユーザー名	任意のユーザー名（例：iotuser）
認証の種類	[パスワード]を選択
パスワード	次の条件を満たす、任意のパスワード （例：!IoTPass1234） * 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）	[無効]を選択
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
場所	任意の場所を選択

2. [仮想マシンのサイズの選択]でサイズを選択します。

（本ハンズオンでは 8GB 以上のメモリが割り当てられるスペックを選択します）

3. [オプション機能の構成]でパブリック受信ポートを選択で"SSH(22)"を選択します。

*** パブリック受信ポートを選択 ⓘ**

SSH (22) ⚠️ ▼

4. [概要]で[作成]ボタンをクリックします。

タスク2：コンテナランタイムのインストール

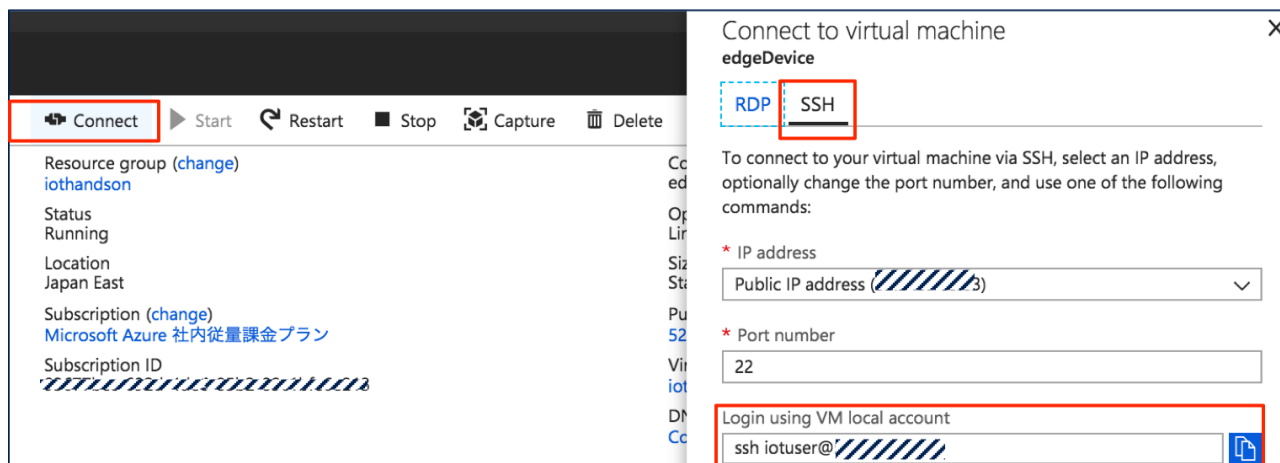
このタスクでは、IoT

Edgeデバイス用にデプロイしたVMにIoT

Edgeのランタイムをインストールします。

1. 接続文字列を確認します。

VMのOverviewを開き[CONNECT]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。

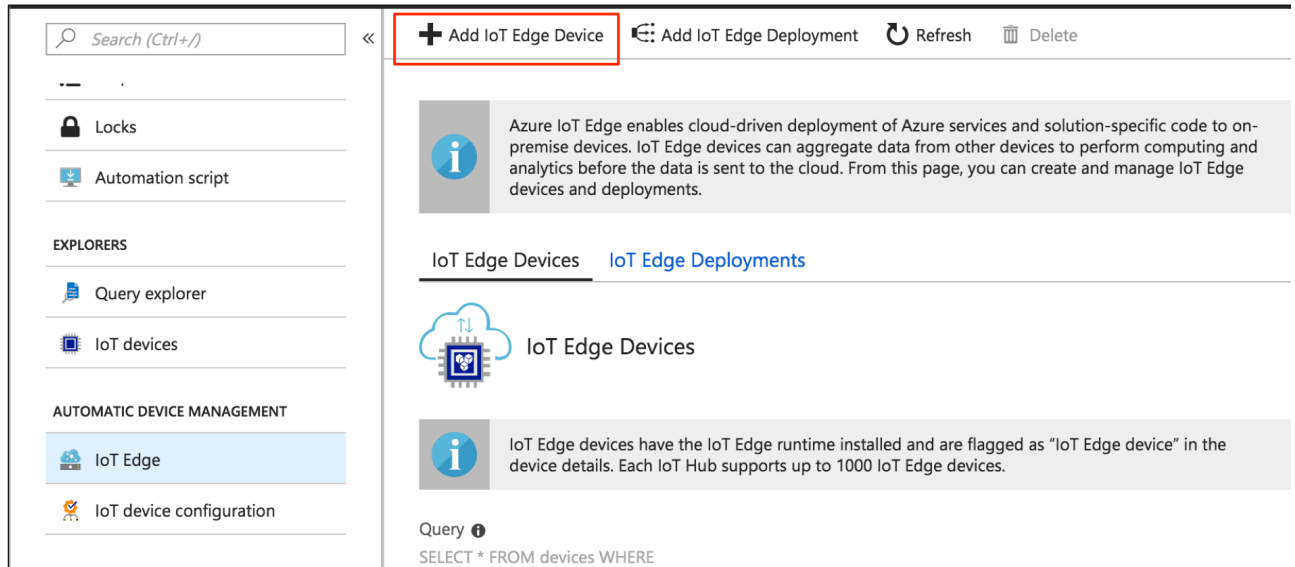
Hello from Docker!

This message shows that your installation appears to be working correctly.

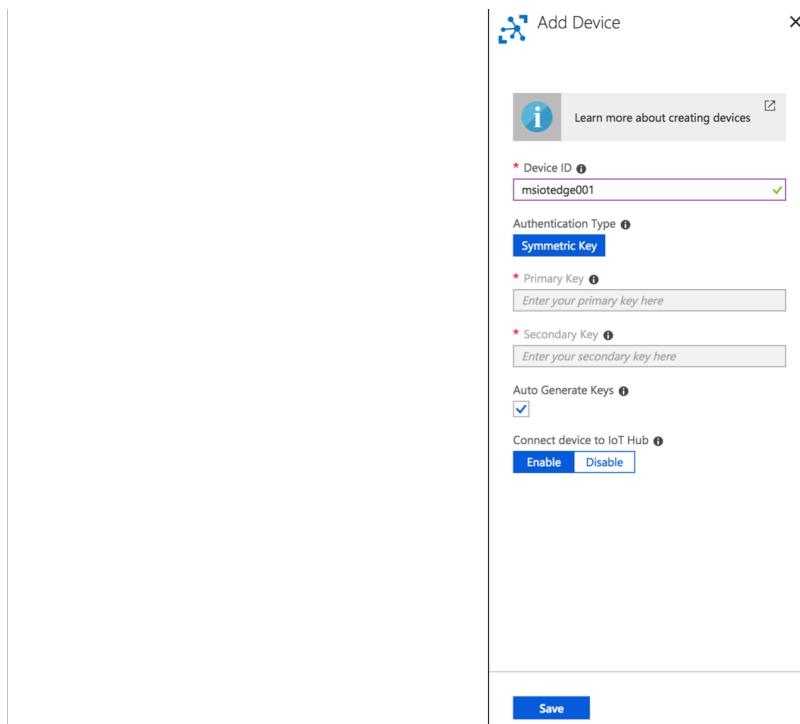
タスク3：IoT Edgeデバイスの追加

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。



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 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 ⓘ

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 ⓘ
msiotedge001

Primary key ⓘ
[Redacted]

Secondary key ⓘ
[Redacted]

Connection string—primary key ⓘ
[Redacted]

Connection string—secondary key ⓘ
[Redacted]

Connect device to IoT Hub ⓘ
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=xxxxxxxxx"
```

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

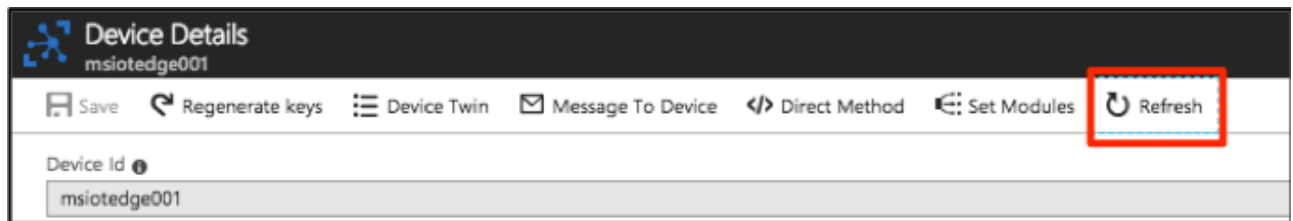
```
systemctl status iotedge
```

7. IoT Edgeランタイムのコンテナログを確認します

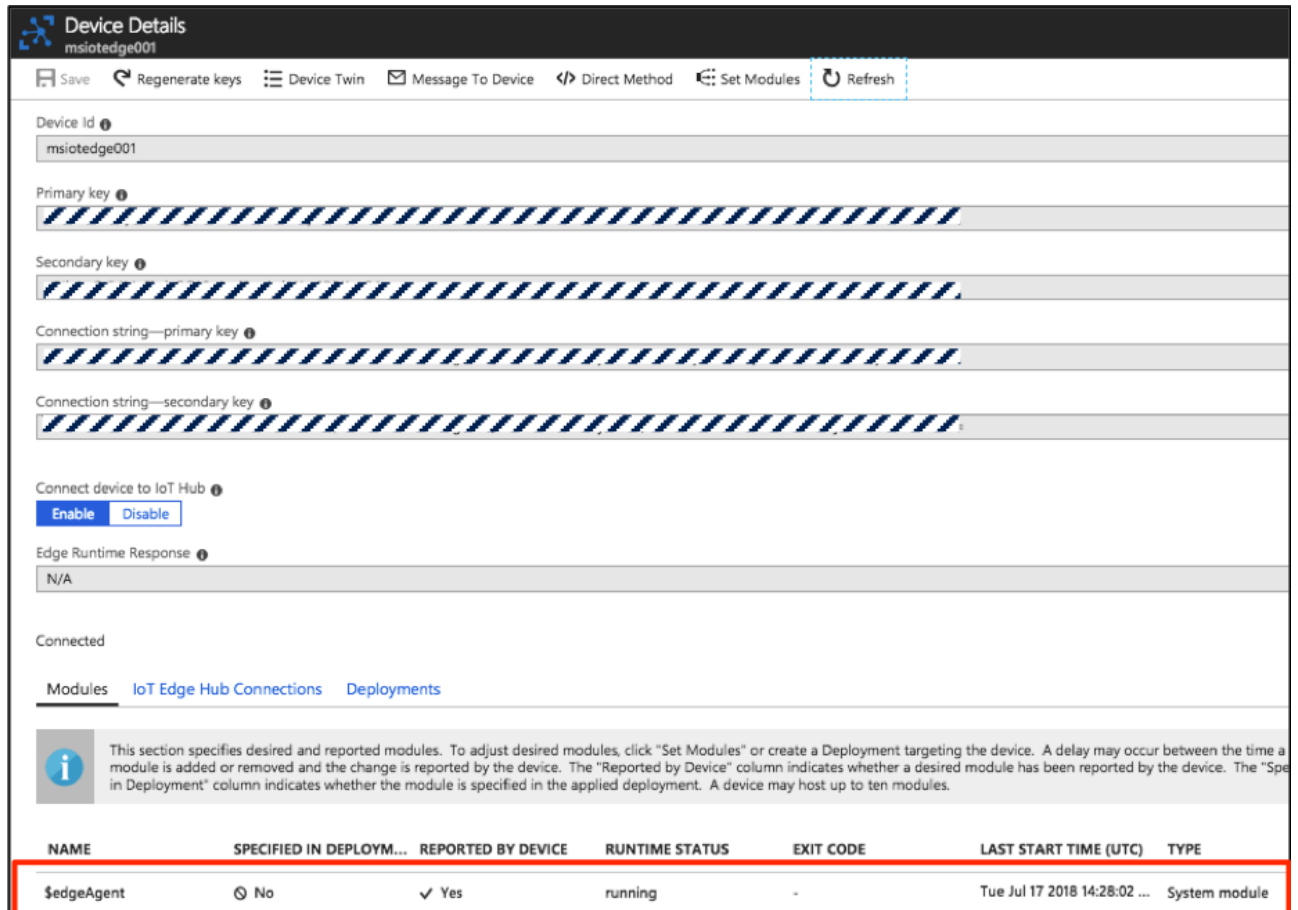
```
docker ps
docker log edgeAgent
```

`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



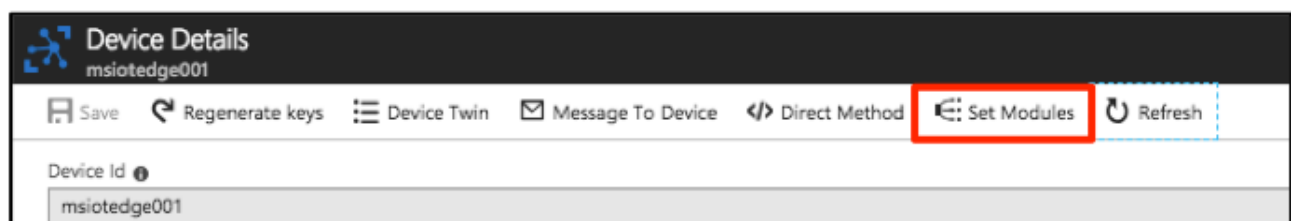
タスク5：IoT

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、擬似的にテレメトリ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

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 500 if it can't find a container registry setting for a module.

Container Registry Settings

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

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.

Deployment Modules

+ Add

🗑 Delete

IoT Edge Module

Azure Stream Analytics Module

Azure Machine Learning Module

	DESIRED STATUS
IoT Edge Module	
Azure Stream Analytics Module	running
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
Set Modules

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

i 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 500 if it can't find a container registry setting for a module.

Container Registry Settings

NAME ADDRESS USER NAME PASSWORD

i 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.

Deployment Modules

+ Add - Delete

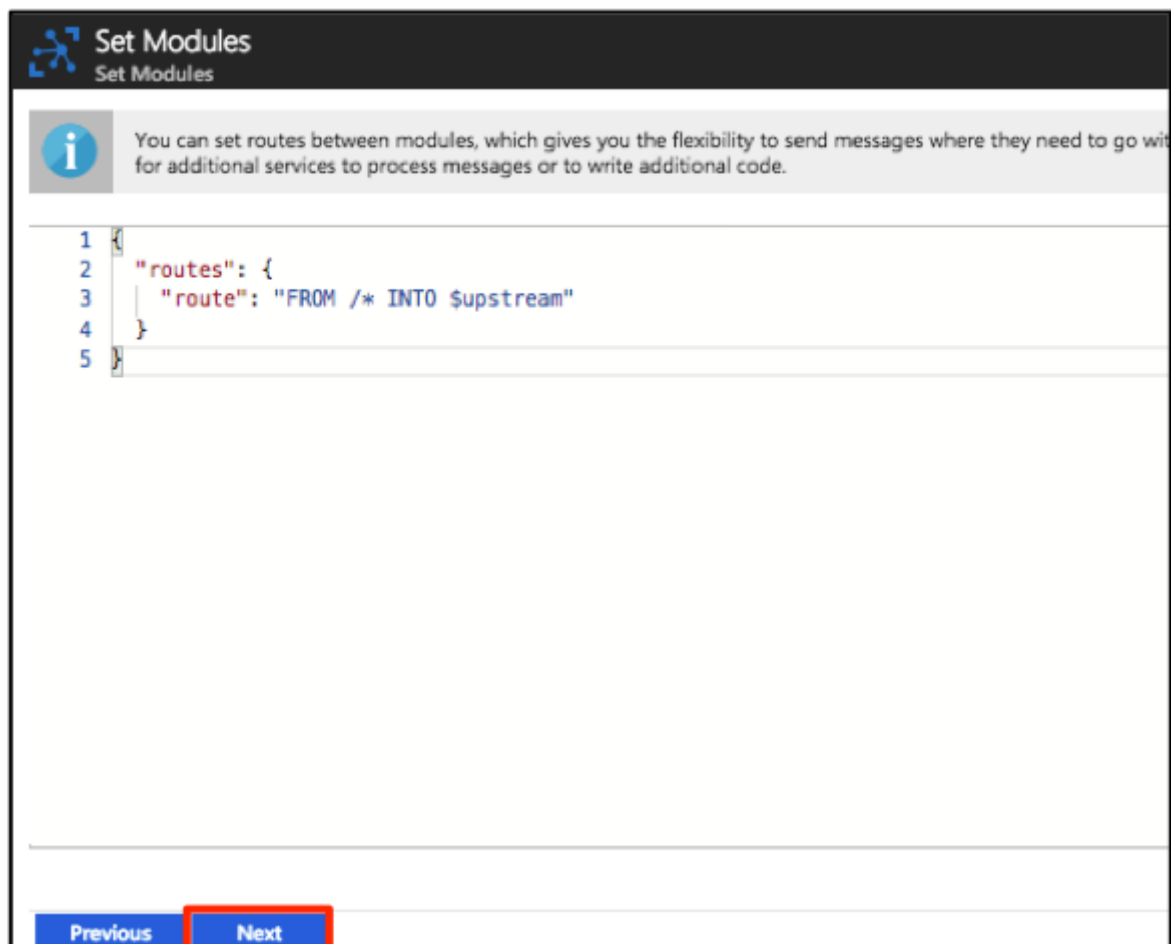
☐ NAME DESIRED STATUS

☐ tempsensor running

Configure advanced Edge Runtime settings

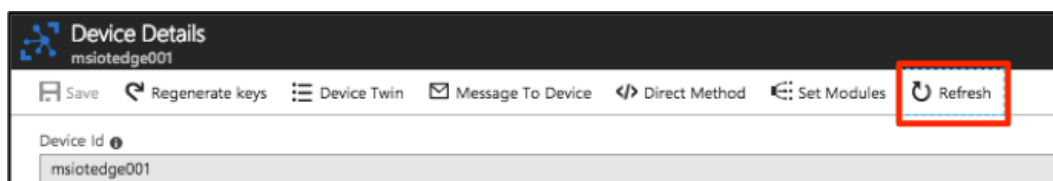
Previous Next Submit

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 /* 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。



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

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、Funcions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/>