

# Azure Percept ワークショップ

日本マイクロソフト株式会社



# 本日の内容

- Azure Percept 概要
- Azure Percept ワークショップ
  - Step1: Percept DK のセットアップ
    - ①サンプルのビジョン AI モデルを試す
    - ②カスタムのビジョン AI モデルを試す
    - (オプション) ③その他のビジョン AI モデルを試す(Open Pose、OCRなど)
  - Step2: Azure 環境のセットアップ
  - Step3: Power BI レポートの作成
  - Step4: Teams アラートの実装

# Azure Percept 概要

# Azure Percept

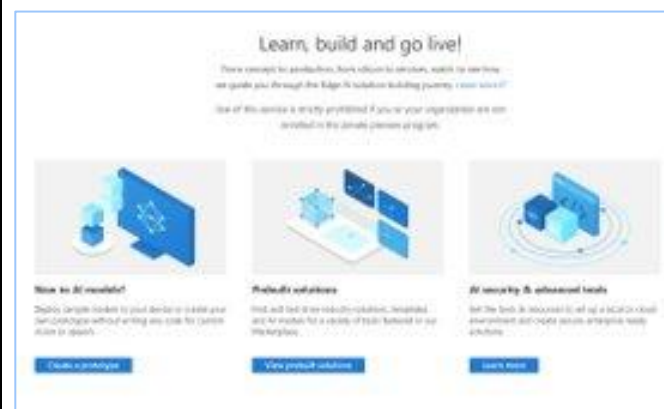
Azure の AI および IoT サービスとシームレスに統合されたハードウェア アクセラレータで、概念実証 (PoC) を数分で開始。

- 1. 包括的なエッジ AI プラットフォームで変革**
  - Azure の AI および IoT サービスと統合されたハードウェア アクセラレータを含む、エンドツーエンドのエッジ AI プラットフォーム。
- 2. エッジ AI ソリューションをシームレスに構築・管理**
  - エッジ AI ライフサイクル管理サービスと事前構築済み AI モデルを利用して、プロトタイプから数分で実稼働へ。
- 3. セキュリティを考慮した設計**
  - マイクロソフトのゼロ トラスト セキュリティ モデルによるハードウェアの "信頼の基点" を利用して、転送中および保存された機密データを保護。



Azure Percept DK

Azure Percept Audio



Azure Percept Studio

# Azure Percept Dev Kit - Vision + Audio 日本販売開始

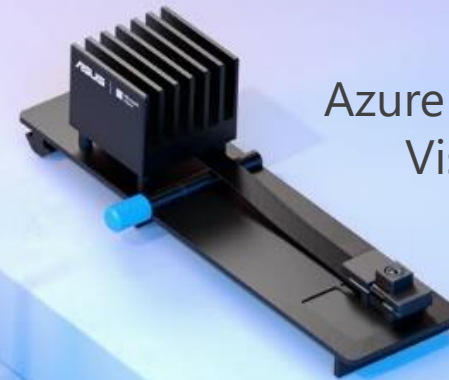
Mics: x4 MEM mics Linear Array  
CODEC: XMOS XUF208  
Security: STM32L462CE  
Audio: 16bit DAC  
Buttons: 2x Momentary  
LEDs: 3x RGB  
Ports: USB Micro B

Azure Percept  
Audio

Processor: NXP iMX8m ARM  
Memory: 4GB  
Storage: 16GB  
Security: TPM 2.0 Nuvoton NCPT750  
Connectivity: WiFi & Bluetooth  
Ports: 1xEthernet  
1x USB C  
2x USB A 3.0



Azure Percept  
Trust Module



Azure Percept  
Vision

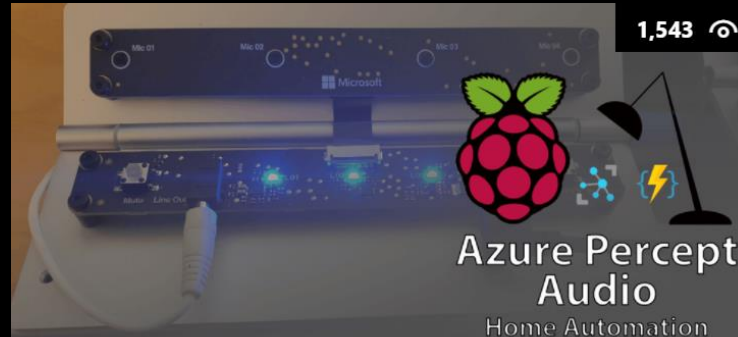
VPU: Intel Movidius Myriad X (MA2085)  
Memory: LPDDR4 2GB  
Sensor: Sony IMX-219 (1.12 um) RGB  
Security: STM32L462CE  
Resolution: 8MP @ 30FPS  
Ports: USB C & MIPI 4 Lane (up to 1.5 Gbps per lane)  
Ports: USB Micro B



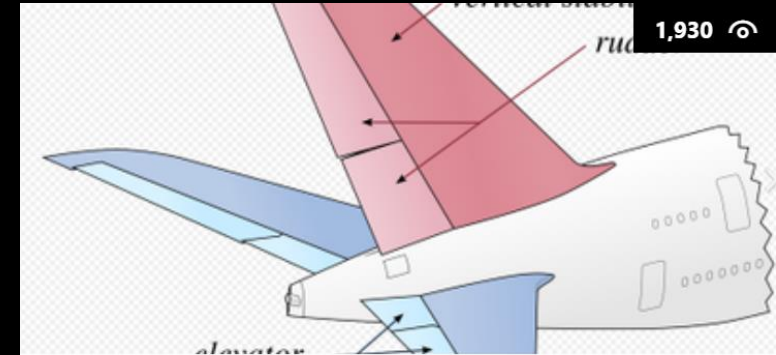
# Real-World Scenarios...



**Building the next smart city with Azure Percept**



**Azure Percept Audio - Home Automation with Azure...**



**Using Azure Percept to build an Aircraft Part Checker**



**Set up your own end-to-end package delivery monitoring AI...**



**Perceptmobile: Azure Percept Obstacle Avoidance LEGO Car**



**Azure Percept – Industrial IoT and Preventive Maintenance**

# 生鮮食品の体積検出



Azure Percept Vision Output

This AI can be trained to recognize any type of fresh produce

- 60% of fresh food spoils and is wasted in retail, representing a \$161.6B/year loss<sup>1</sup>
- Video AI can provide the ground truth for analytics containing the purchasing and restocking rates of perishables at retail.
- These purchasing analytics enables a more efficient supply chain, resulting in less CO2 emissions from transportation of fresh food and less fertilizer in the soils.
- This creates a healthier planet.

<sup>1</sup> Source: USDA



# AI For Good: Percept による 絶滅危惧種 (鳥) カウンター

Bird feeder with integrated Percept AI camera



Percept AI output with Chickadee (*Poecile gambeli*) detection

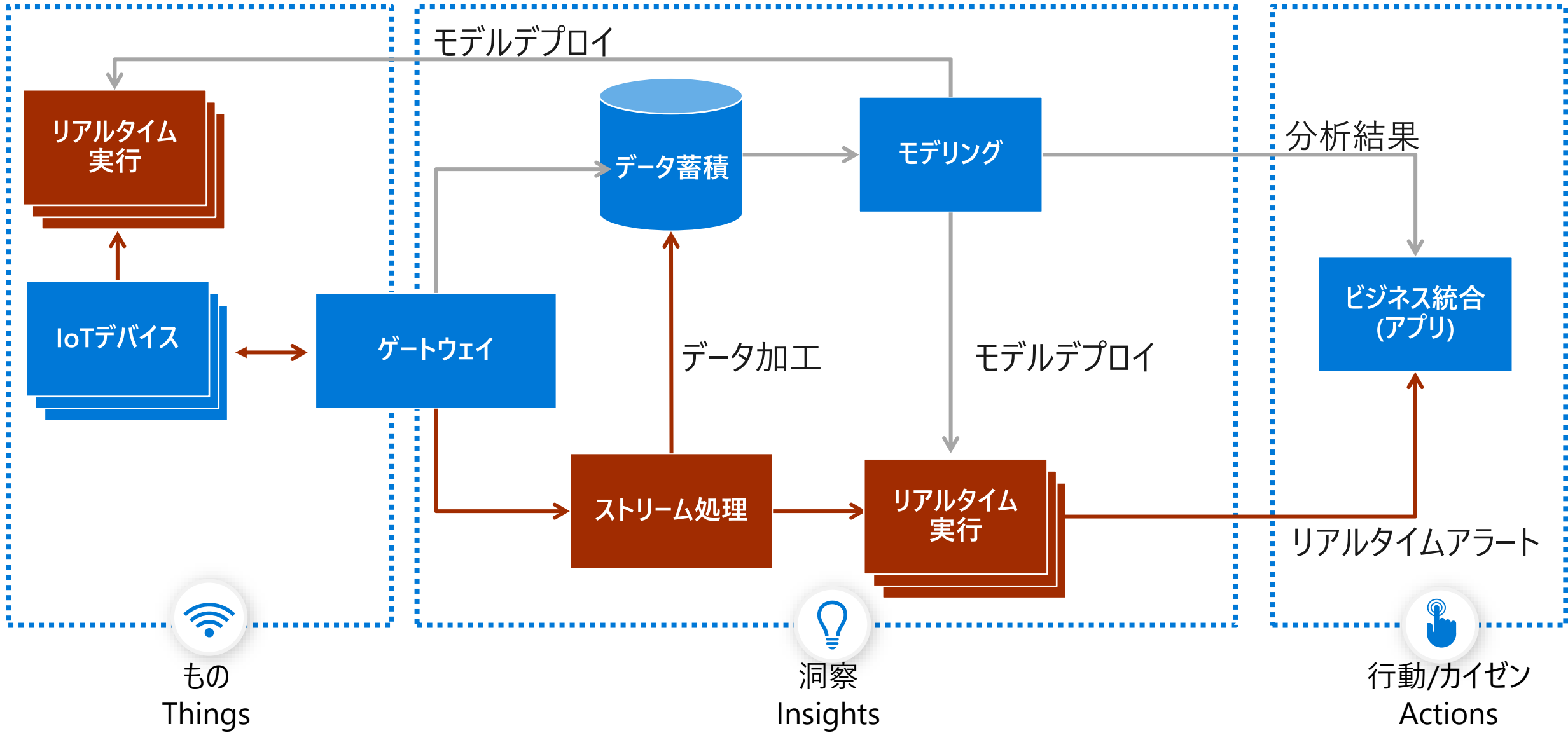




# Azure Percept ワークショップ

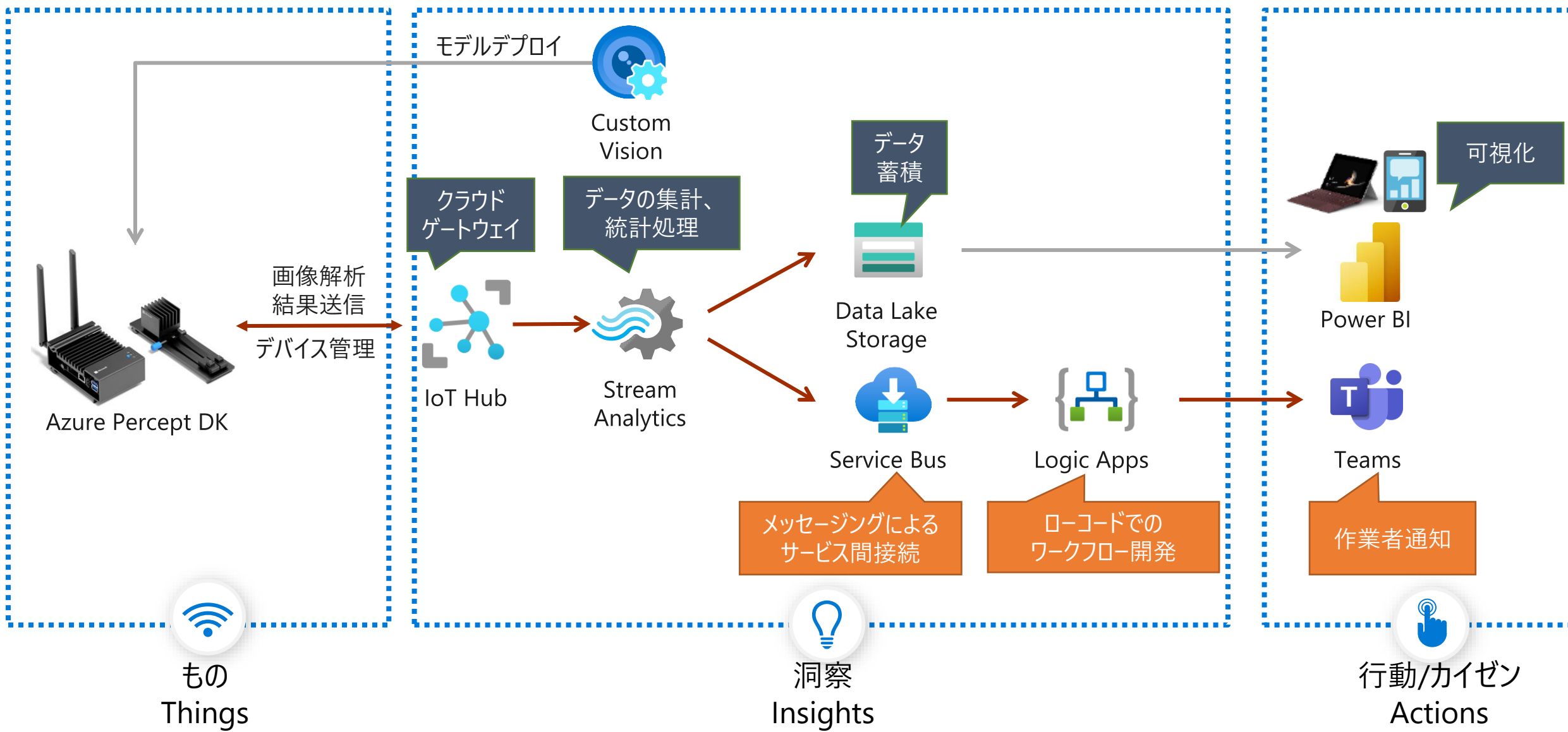
# AI + IoTリファレンスアーキテクチャ

→ バッチ  
→ リアルタイム



# ハンズオン構成

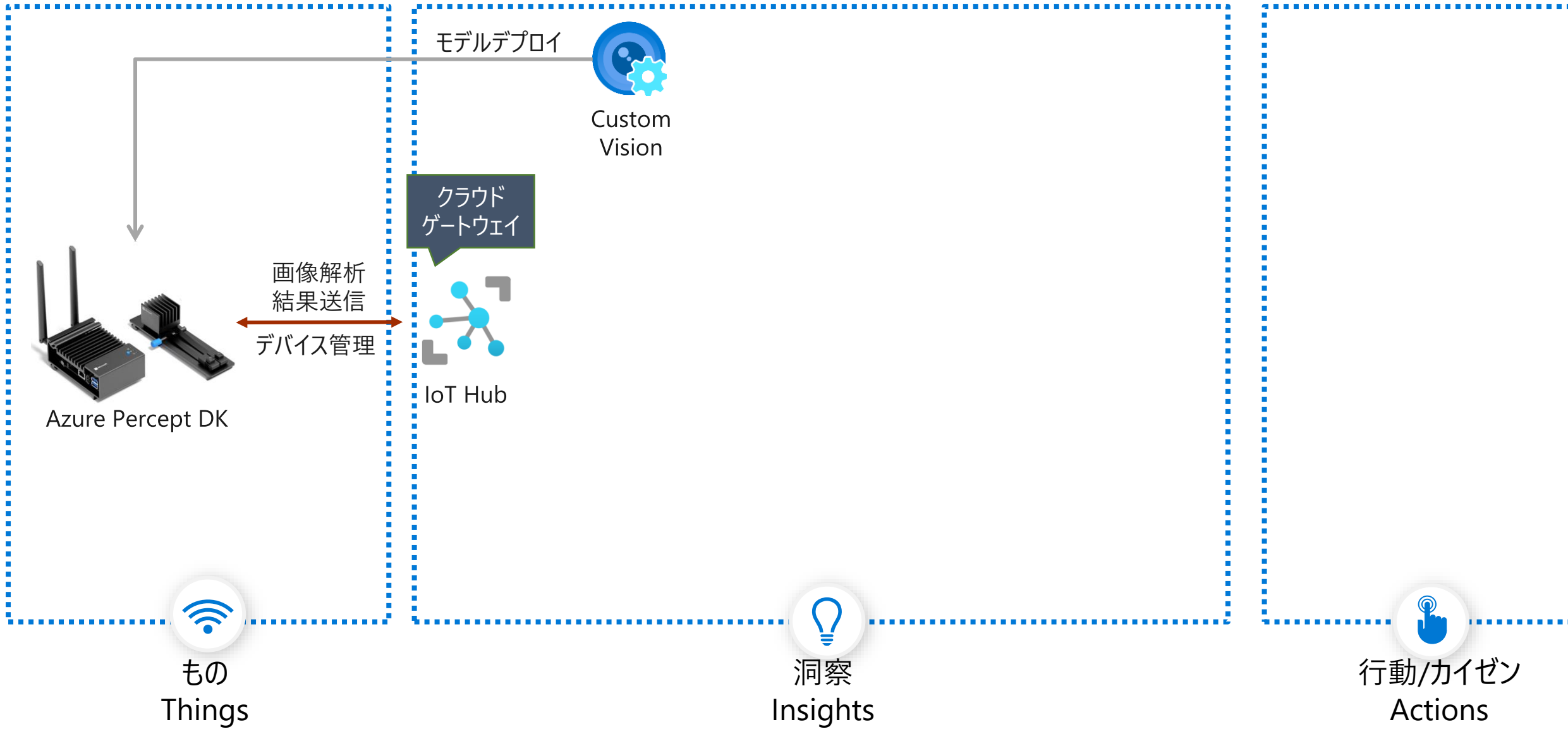
→ バッチ  
→ リアルタイム





# Step1: Percept DK のセットアップ

→ バッチ  
→ リアルタイム



# Percept DK のセットアップ

- 以下手順に従って、セットアップを実施します
  - [Azure Percept DK デバイスを箱から取り出して組み立てる | Microsoft Docs](#)
  - [Azure Percept DK デバイスを設定する | Microsoft Docs](#)
- ※はじめて使用する場合、以下手順にしたがって最新のOSとファームウェアの更新を実施することをお勧めします
  - [USB-C ケーブル接続で Azure Percept DK を更新する | Microsoft Docs](#)

# ① サンプルのビジョン AI モデルを試す

- 以下手順に従って、サンプルのビジョン AI モデルをデプロイします
  - [Azure Percept DK の RTSP ビデオ ストリームを表示する | Microsoft Docs](#)
  - [Azure Percept DK にビジョン AI モデルをデプロイする | Microsoft Docs](#)
- 利用可能なサンプルモデル
  - 人物検出
  - 車両検出
  - 一般的な物体検出
  - 棚の製品検出



# ①サンプルのビジョン AI モデルを試す

## Vision モデルのサンプルを試す

すぐにデバイスにデプロイする

×

すぐに開始するには、使用可能なサンプルのいずれかを選択します。該当のタイルをクリックしてください。

サンプル モデル ①

### ☐ 一般的なオブジェクトの検出

共通オブジェクトを識別します



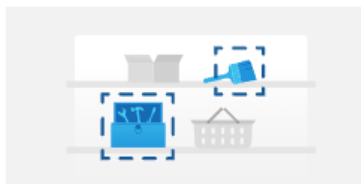
### ☐ 人物検出

ライブ ストリームで人物を識別します



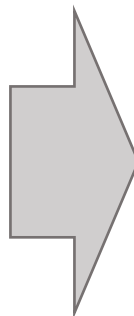
### ☐ 棚の製品

製品が棚にない場合に製品を識別します

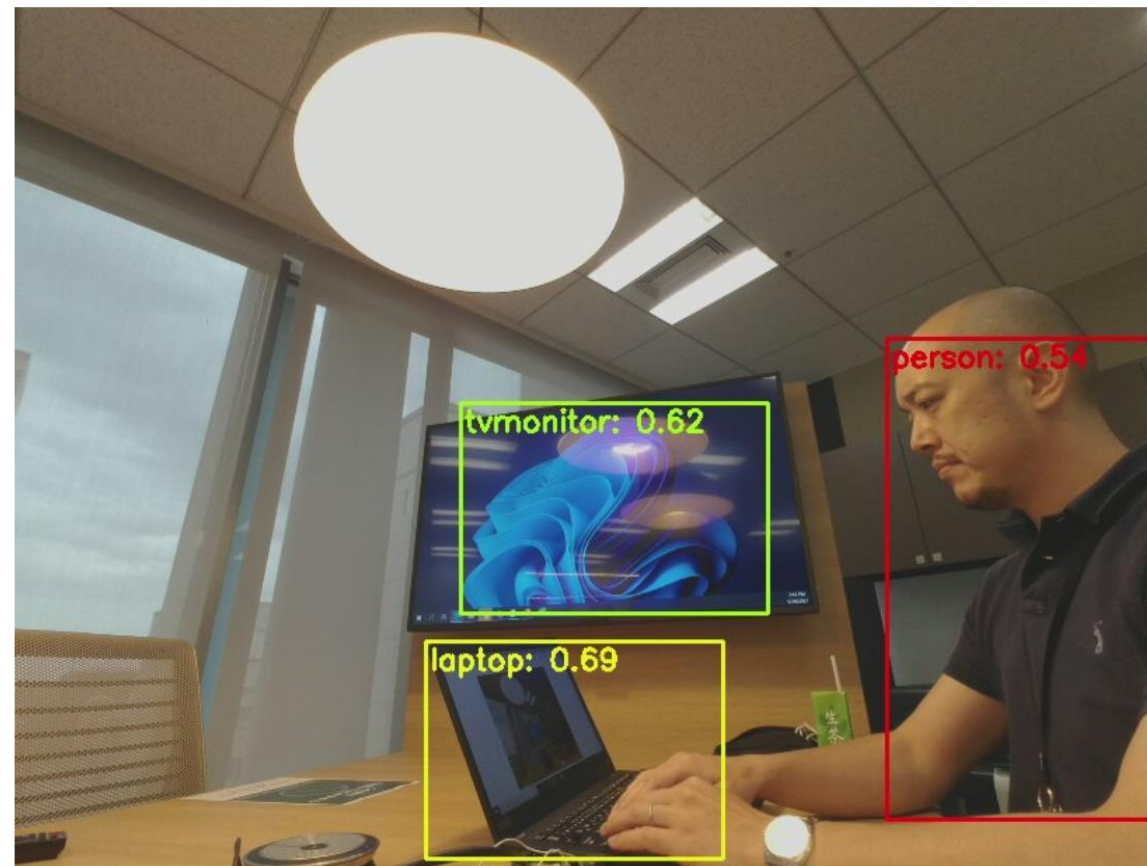


### ☐ 車両検出

車両の種類の識別



## Webstream Video

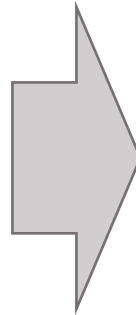
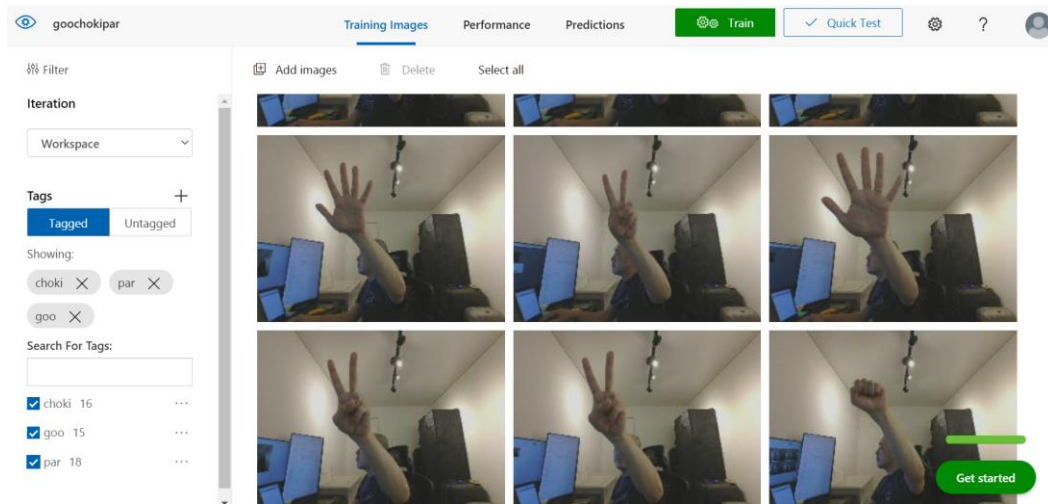


## ②カスタムのビジョン AI モデルを試す

- 以下手順に従って、カスタムのビジョン AI モデルを作成およびデプロイします
  - [Azure Percept Studio でコードなしのビジョン ソリューションを作成する | Microsoft Docs](#)
- Custom Vision による以下モデルが利用可能
  - 分類 (Classification)
  - 物体検出 (Object Detection)

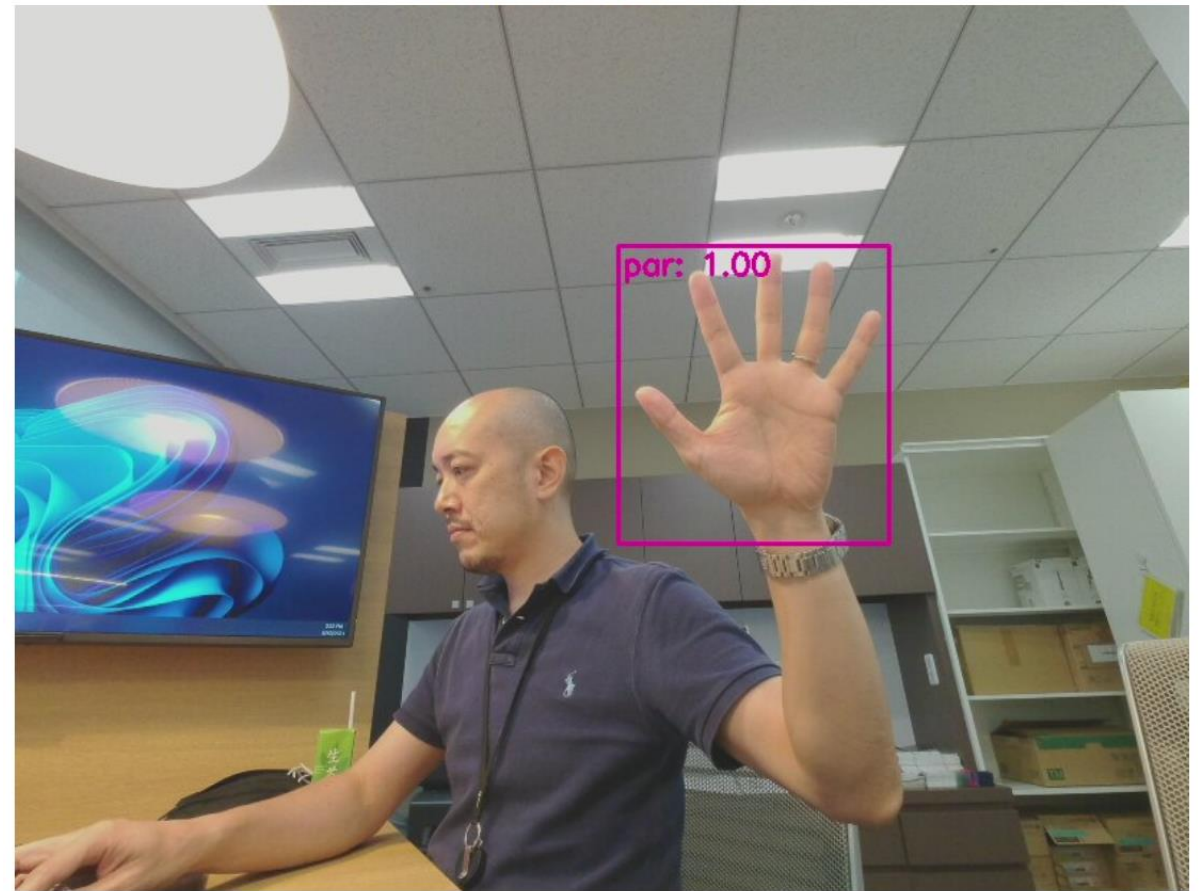
## ②カスタムのビジョン AI モデルを試す

Custom Vision でカスタム AI モデルを学習



Percept にモデルをデプロイ

**Webstream Video**





# ③その他のビジョン AI モデルを試す

- 以下を参考に、Open Pose のモデルをデプロイします
  - [Azure Percept を使用した高度な開発 | Microsoft Docs](#)
  - [GitHub - microsoft/azure-percept-advanced-development: Azure Percept DK advanced topics](#)

Model	Source	License	URL
Open Pose	<a href="#">Intel Open Model Zoo</a>	Apache 2.0	<a href="https://aedsamples.blob.core.windows.net/vision/aeddevkitnew/openpose.zip">https://aedsamples.blob.core.windows.net/vision/aeddevkitnew/openpose.zip</a>

- 上記 URL を azureeyemodule のモジュールツイン の ModelZipUrl に指定します
  - Azure Portal 上の IoT Hub の画面における [IoT Edge] > [該当デバイスID] > [モジュールを設定する] > [azureeyemodule] > [モジュールツインの設定] より更新

## IoT Edge モジュールの更新

IoT Edge カスタム モジュールの設定を指定してください。

[詳細](#)

IoT Edge モジュール名 \*

azureeyemodule

モジュールの設定   環境変数   コンテナの作成オプション   モジュール ツインの設定

指定のモジュール ツイン プロパティが更新されて、入力した JSON が反映されます。

[デプロイの詳細](#)

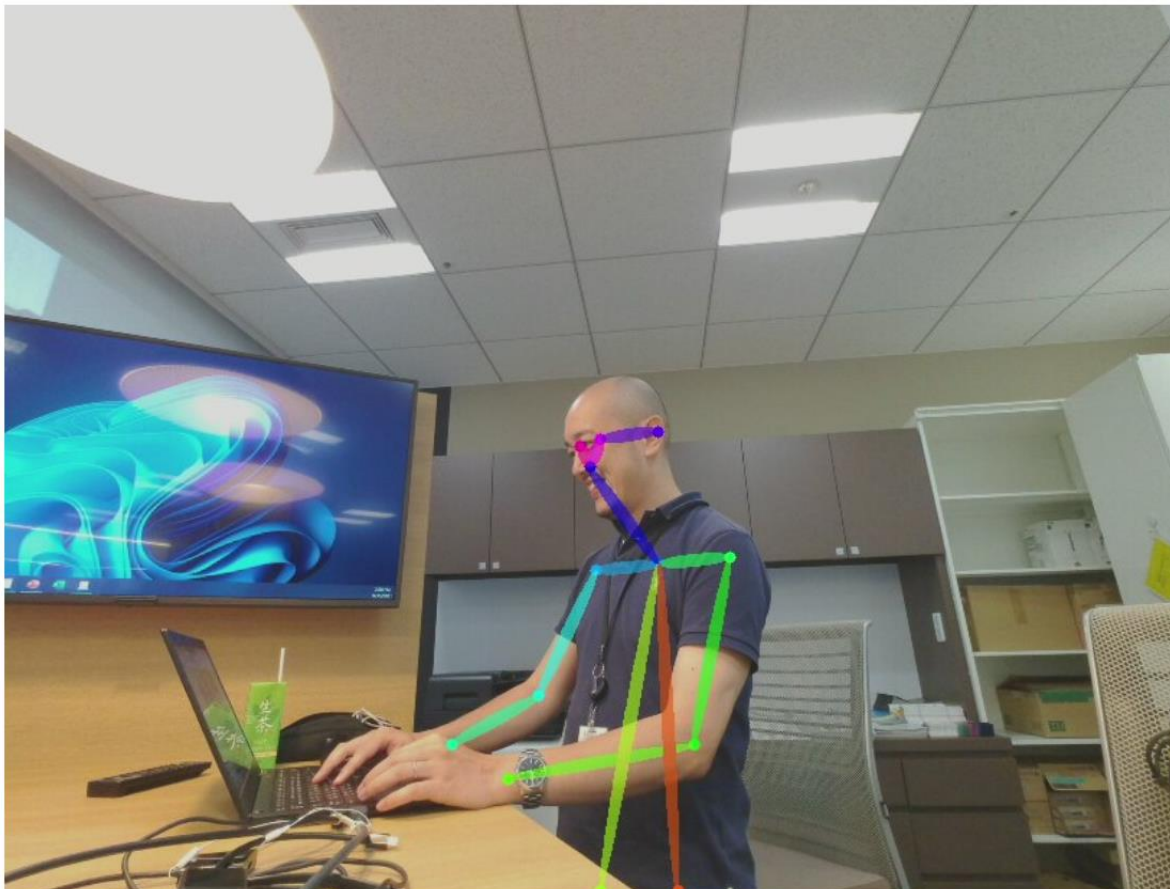
```
1 {
2   "Logging": true,
3   "ModelZipUrl": "https://aedsamples.blob.core.windows.net/vision/aeddevkitnew/openpose.zip",
4   "RawStream": true,
5   "ResultStream": true,
6   "Running": true,
7   "TelemetryInterval": 10,
8   "TelemetryIntervalNeuralNetworkMs": 10000
9 }
```

- 利用可能なモデル
  - Faster RCNN ResNet 50
  - Open Pose
  - Optical Character Recognition
  - Person Detection
  - Product Detection
  - SSD General
  - Tiny YOLOv2 General
  - Unet for Semantic Segmentation
  - Vehicle Detection

### ③その他のビジョン AI モデルを試す

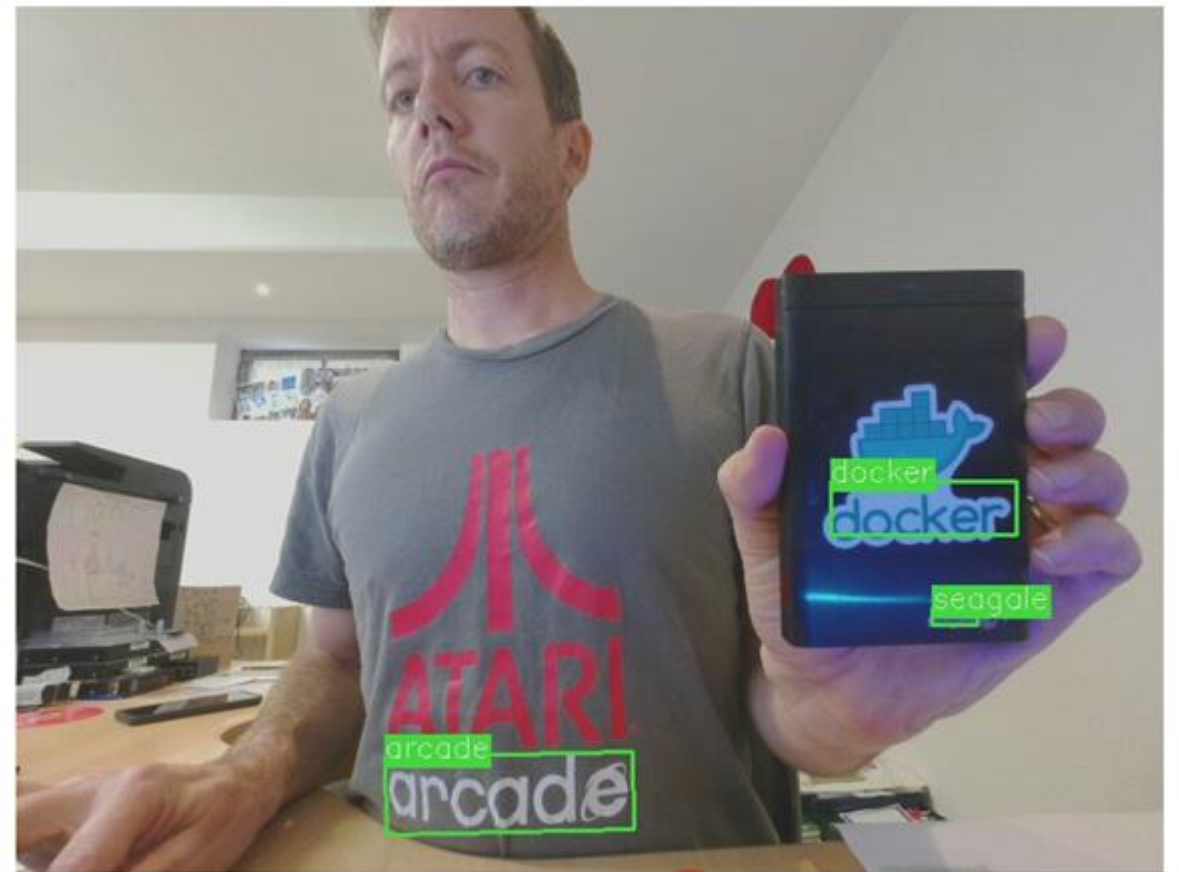
例：Open Pose

**Webstream Video**



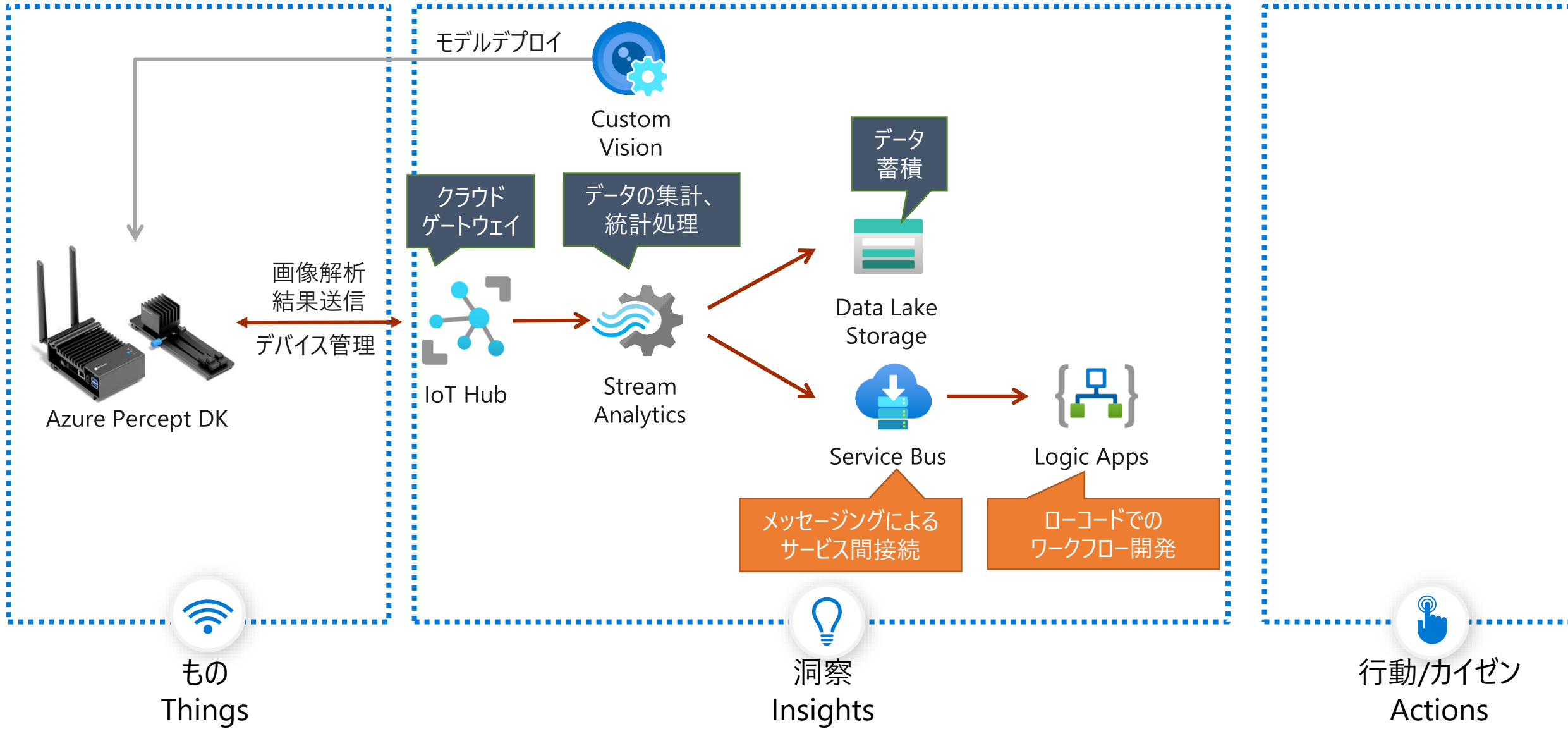
例：OCR

**Webstream Video**




# Step2: Azure 環境のセットアップ

→ バッチ  
→ リアルタイム



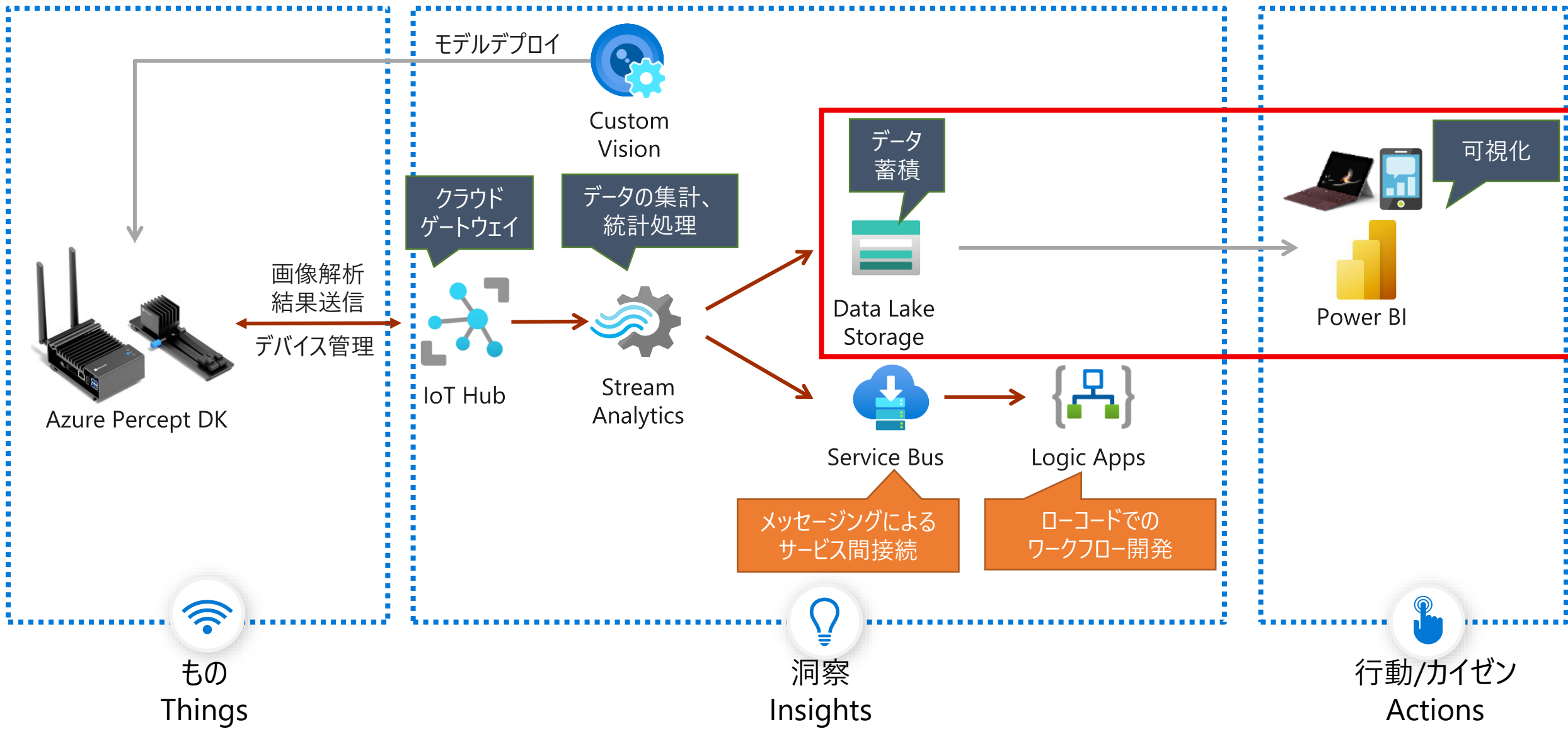
# Azure 環境のセットアップ

- 以下リンク先の  をクリックし、Azure 関連リソースをデプロイします
  - [GitHub - yahanda/hands-on-percept](https://github.com/yahanda/hands-on-percept)



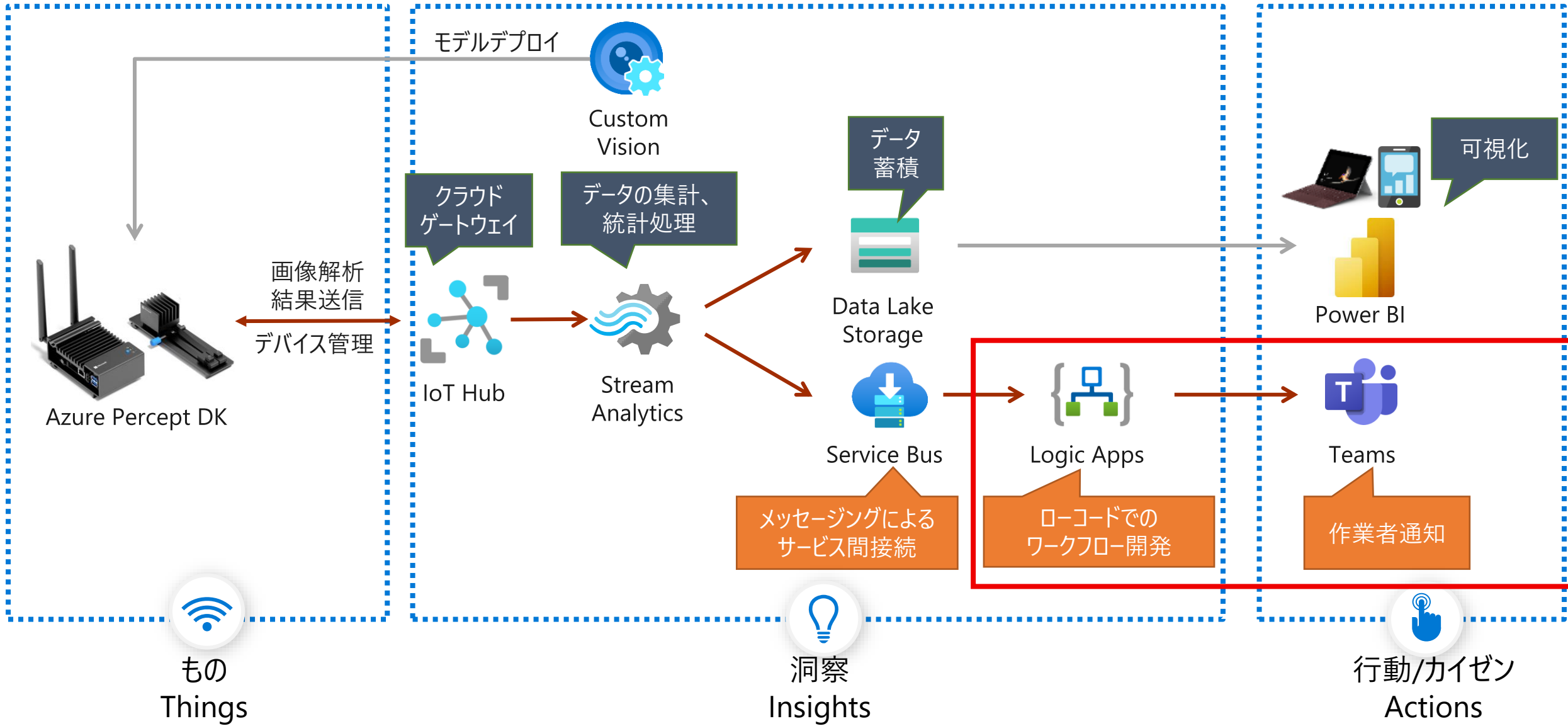
# Step3: Power BI レポートの作成

→ バッチ  
→ リアルタイム



# Step4: Teams アラートの実装

→ バッチ  
→ リアルタイム



# Logic App のトリガーの構成

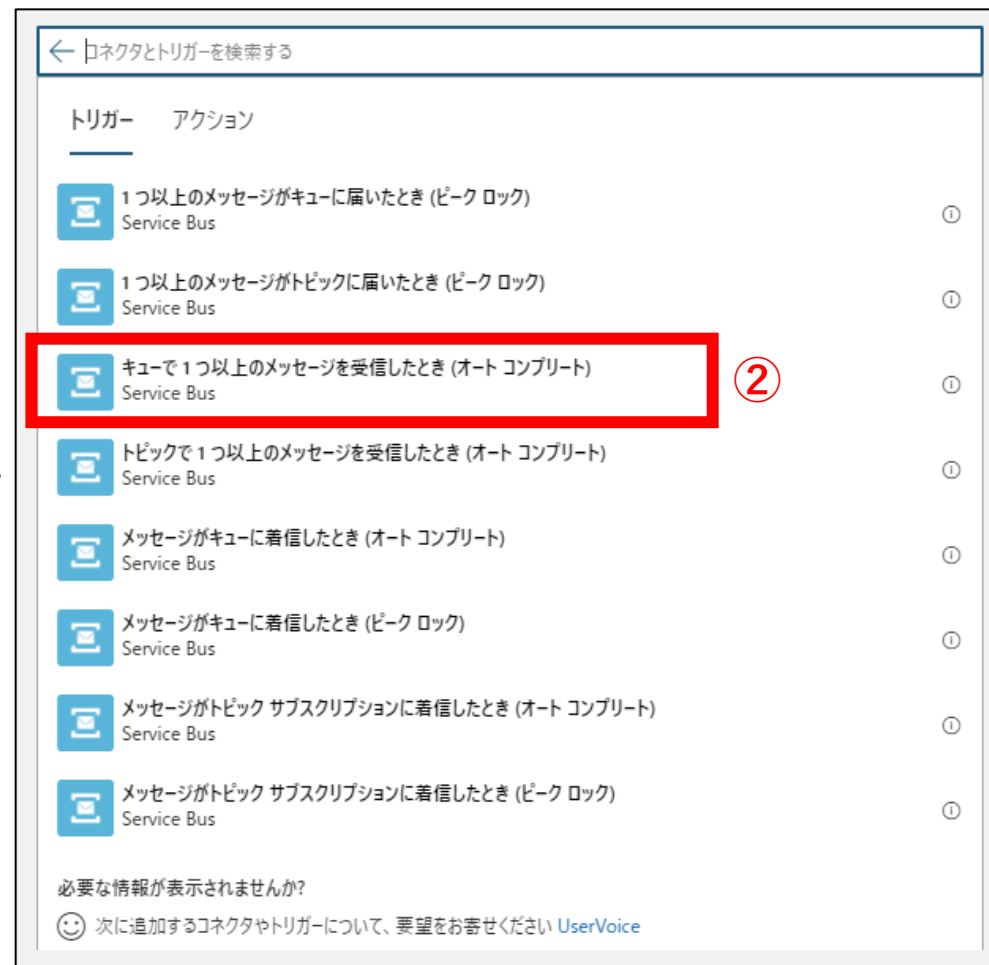
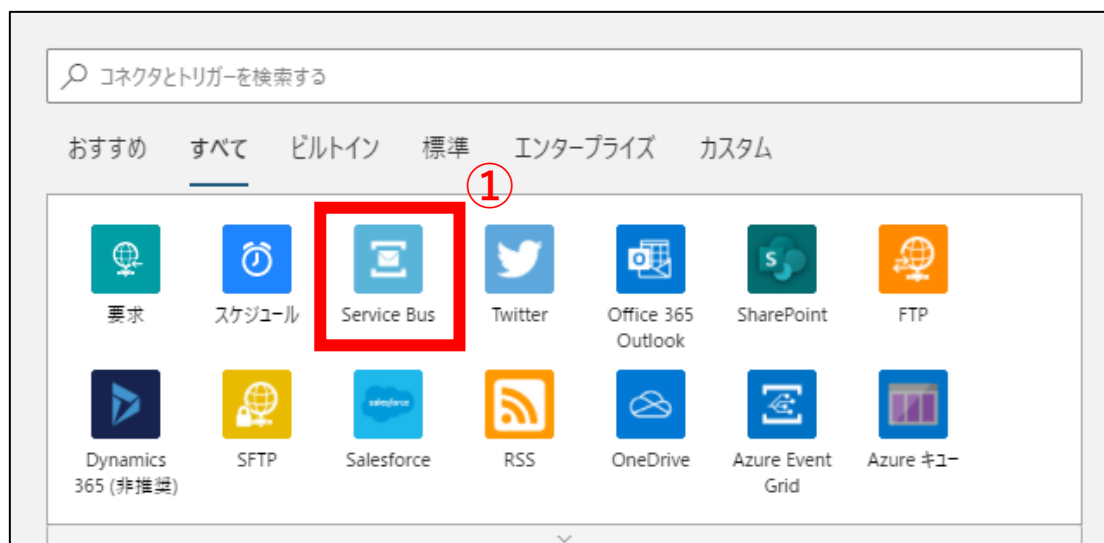
Logic Appの画面にて、以下の順でボタンを押下します。

- ① 開発ツールにて[ロジック アプリ デザイナー]を選択
- ② テンプレートまで下へスクロールし、[空のロジック アプリ] を選択




# Logic App のトリガーの構成

- ① [すべて] タブを選択して、[Service Bus]を選択
- ② [トリガー]で、[キューで1つ以上のメッセージを受信したとき (オート コンプリート)]を選択



# Logic App のトリガーの構成

- ① [キュー名] で、ドロップダウンから先の手順で作成したキューを選択
- ② [項目を確認する頻度] を3秒に設定
- ③ [+新しいステップ] を選択



保存 破棄 トリガーの実行 デザイナー コードビュー パラメーター テンプレート コネクタ ヘルプ 情報

キューで1つ以上のメッセージを受信したとき (オート コンプリート)

\*キュー名

alert-queue ①

最大メッセージ数

20

キューの種類

Main

項目を確認する頻度

3 秒 ②

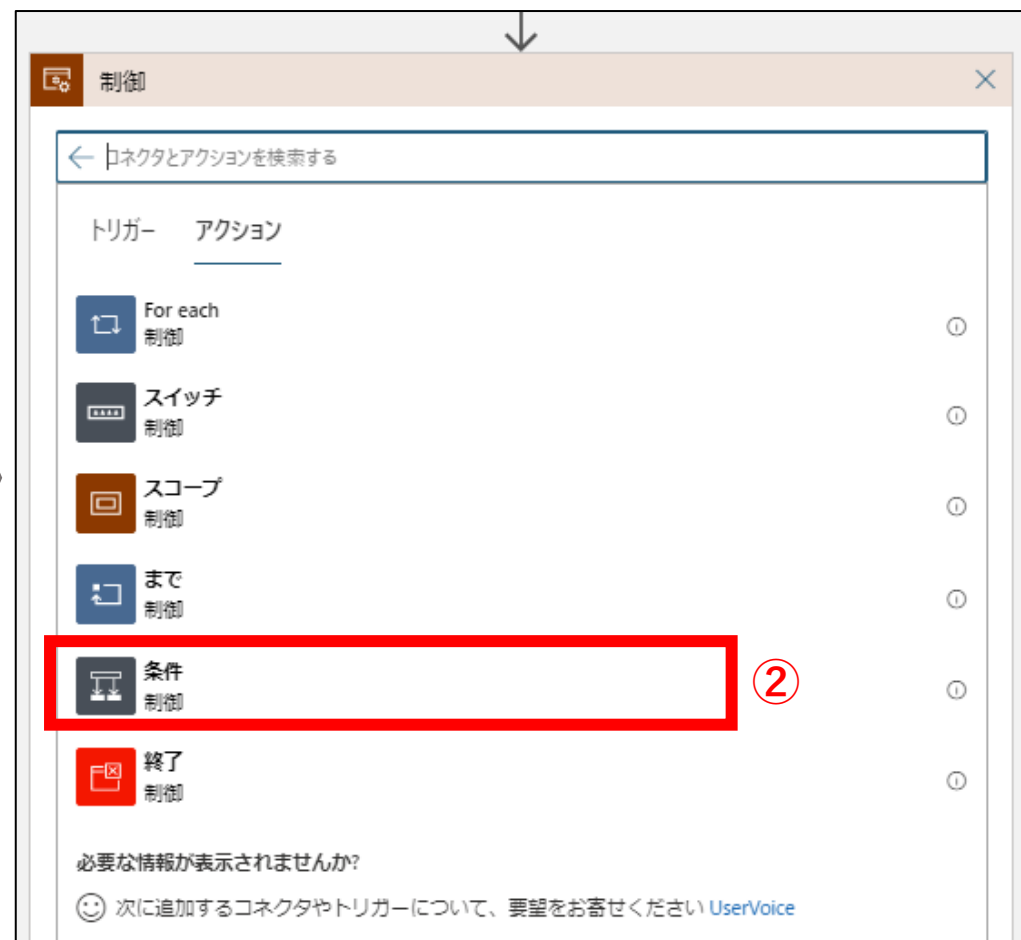
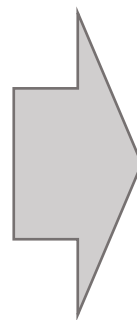
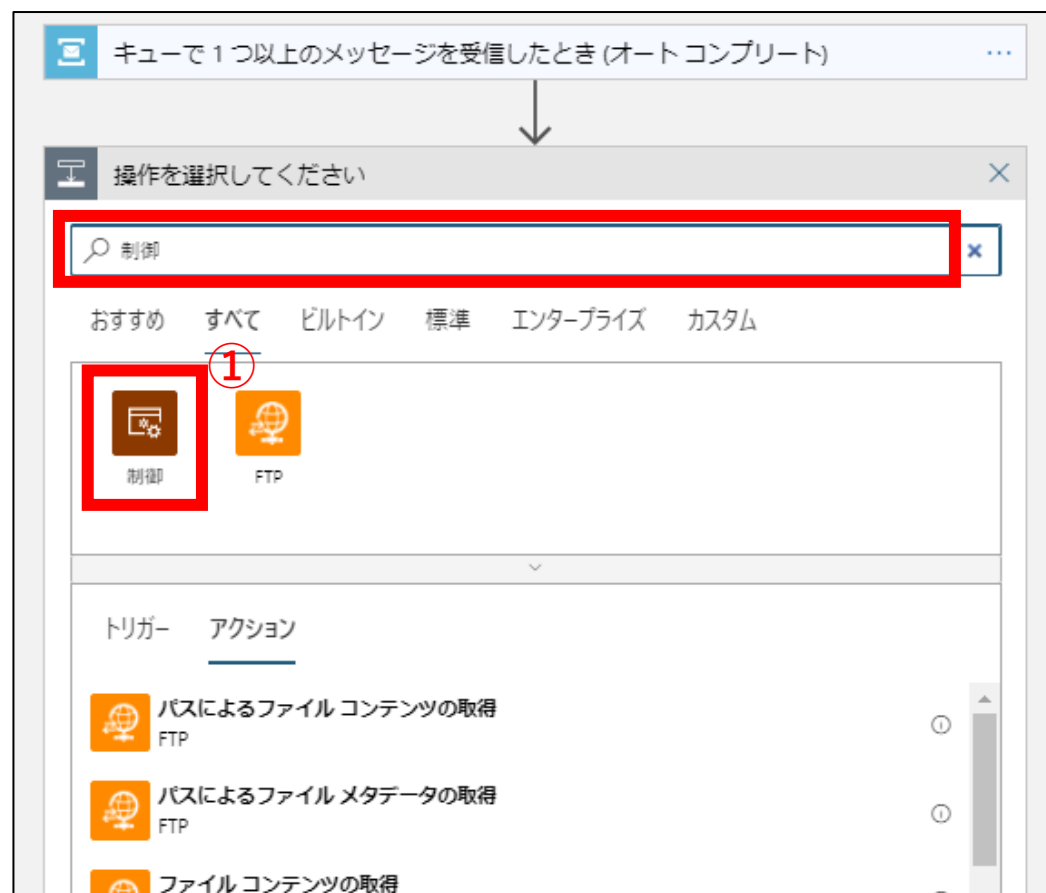
Add new parameter

sb-handa-handson に接続しました。 接続を変更してください。

+ 新しいステップ ③

# Logic App のアクションの構成

- ① [すべて] タブを選択して、検索ボックスに[制御]と入力し、[制御]を選択
- ② [アクション]で、[条件]を選択





# Logic App のアクションの構成

- ① [条件] にて [値の選択] を選択
- ② [式] タブを選択して、以下式を入力  
`json(base64ToString(triggerBody()?['ContentData'])).label`
- ③ [OK] を選択

Stream Analyticsで出力した label を取得

キューで1つ以上のメッセージを受信したとき (オートコンプリート)

条件

And ▾

① 値の選択

動的なコンテンツの追加 +

+ 追加 ▾

次の値に等しい ▾ 値の選択 ...

値へのアクセスや、値の変換および比較といった基本的な処理を行う式を追加します。動的コンテンツの詳細情報。

動的なコンテンツ ② 式

`json(base64ToString(triggerBody()?['ContentData'])).label`

OK ③

文字列関数 もっと見る

True

アクションの追加

# Logic App のアクションの構成

- ① ドロップダウンより、[次の値に等しい] を選択
- ② 該当ラベル名(例: laptop) を入力 ※アラート上げる際のラベル名をここで指定しています。
- ③ True 内の [アクションの追加] を選択

The screenshot displays the Logic App Designer interface. At the top, a navigation bar includes icons for saving, discarding, running the trigger, switching between Designer and Code view, parameters, templates, connectors, help, and information. The main workspace shows a workflow starting with a trigger 'When a message is received in a queue (auto-complete)'. Below the trigger is a condition action labeled '条件' (Condition). The condition is configured with the operator 'And' and a single condition: 'json(...) 次の値に等しい' (json(...) equals the next value). The value 'laptop' is entered in the adjacent text box. Red annotations highlight the configuration steps: ① points to the operator dropdown, ② points to the value input box, and ③ points to the 'アクションの追加' (Add Action) button in the 'True' branch. The 'True' branch is highlighted in green, and the 'False' branch is highlighted in red.

保存 破棄 トリガーの実行 デザイナー コード ビュー パラメーター テンプレート コネクタ ヘルプ 情報

When a message is received in a queue (auto-complete)

条件

And

json(...) 次の値に等しい laptop

+ 追加

True

アクションの追加 ③

False

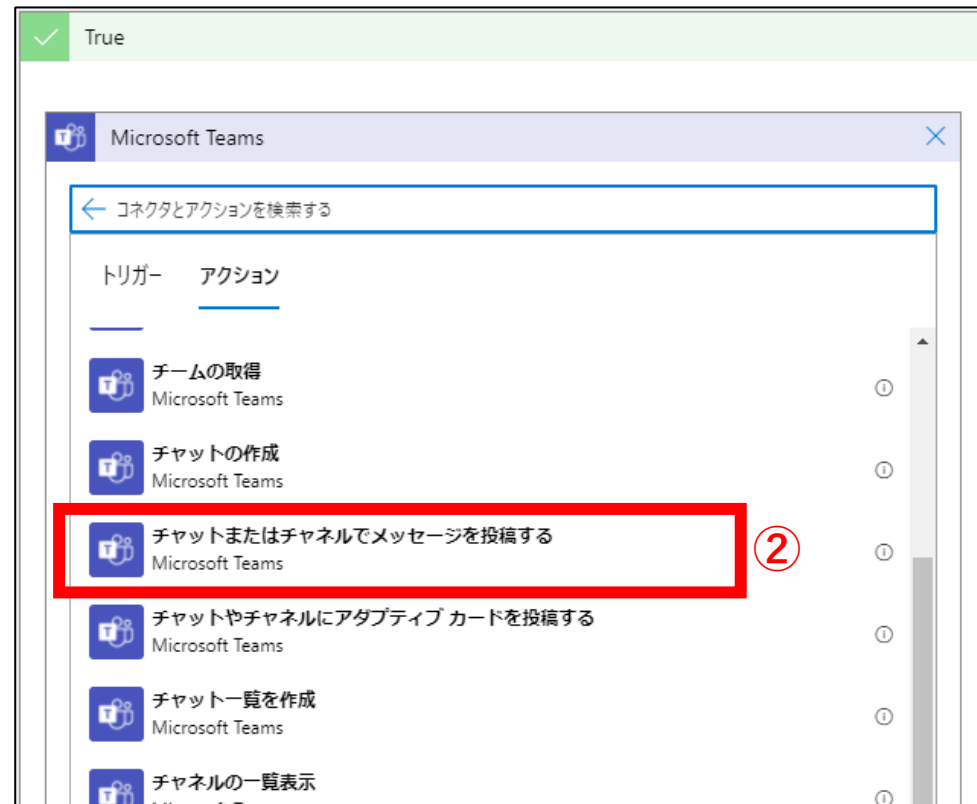
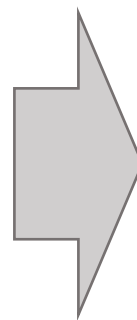
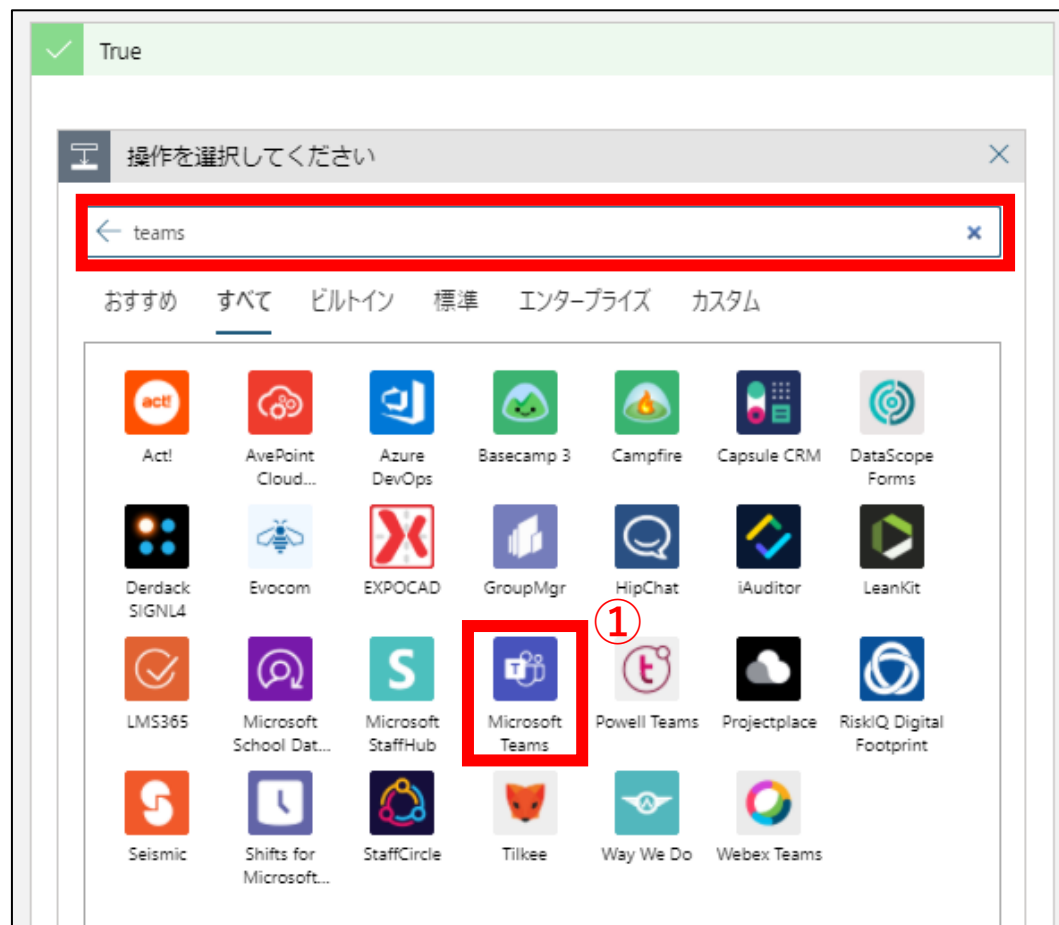
アクションの追加

# Logic App のアクションの構成

① [すべて] タブを選択して、検索ボックスに[teams]と入力し、[Microsoft Teams]を選択

② [アクション]で、[チャットまたはチャンネルでメッセージを投稿する]を選択

※サインイン画面が出たら、ご自身のTeams ID にてサインインします



# Logic App のアクションの構成

✓ True

チャットまたはチャンネルでメッセージを投稿する

\* 投稿者 フロー ボット ① ✓

\* 投稿先 Channel ② ✓

\* Team MyTeam ③ ✓

\* Channel alert sample ④ ✓

\* Message

フォント ▼ 12 ▼ **B** *I* U [List Icons] [Link Icon] [Image Icon]

laptop が検出されました ⑤

内容 ×

yahanda@microsoft.com に接続しました。 [接続を変更してください。](#)

以下の項目を、適宜設定ください。

① 投稿者： [Flow bot] を選択

② 投稿先： [Channel] を選択

③ Team：

④ Channel：

⑤ Message：

※[動的なコンテンツの追加] > [内容] より Stream

Analytics から送信されたメッセージのコンテンツが追加できます

# Logic App のアクションの構成

[保存]を選択して、変更内容を保存

The screenshot displays the Microsoft Logic App Designer interface. At the top, a navigation bar includes icons and labels for '保存' (Save), '破棄' (Discard), 'トリガーの実行' (Run Trigger), 'デザイナー' (Designer), 'コードビュー' (Code View), 'パラメーター' (Parameters), 'テンプレート' (Templates), 'コネクタ' (Connectors), 'ヘルプ' (Help), and '情報' (Info). The '保存' button is highlighted with a red box.

The main workspace shows a workflow starting with a trigger: 'When a message is received in a queue (auto-complete)'. This is followed by a condition step labeled '条件' (Condition). The condition is configured with the following settings:

- Operator: And
- Condition 1: json(...) x (selected)
- Comparison: 次の値に等しい (Equal to the next value)
- Value: laptop

Below the condition, there are two paths for the workflow:

- True Path (Green header):** Labeled 'チャットまたはチャネルでメッセージを投稿する' (Post message in chat or channel). It includes fields for:
  - 投稿者 (Post author): フロー ボット (Flow bot)
  - 投稿先 (Post to): Channel
  - Team: MyTeam
  - Channel: alert sample
  - Message: A text area containing 'laptop が検出されました' (laptop detected) with a rich text editor toolbar above it.
- False Path (Red header):** Labeled 'アクションの追加' (Add action).

At the bottom of the True path, there is a status message: 'yahanda@microsoft.com に接続しました。接続を変更してください。' (Connected to yahanda@microsoft.com. Please change the connection.) and another 'アクションの追加' (Add action) button.