

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: Azure環境のセットアップ
 - Step2: Percept DK のセットアップ
(組み立て、設定、サンプルを試す)
 - Step3: Stream Analyticsの起動
 - Step4: Power BI レポートの作成
 - Step5: Teams アラートの実装
(Service bus、Logic Apps)

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。

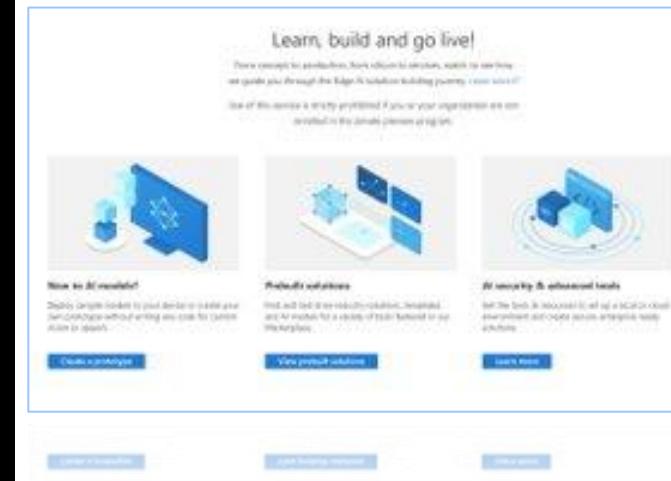
3 セキュリティを考慮した設計

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



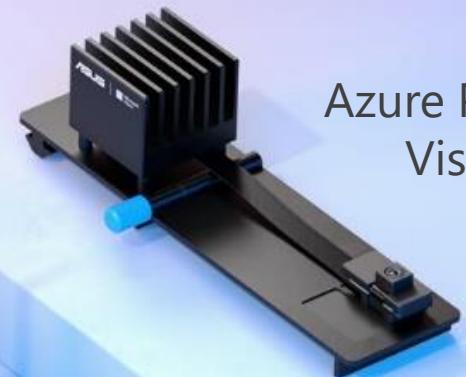
Processor: NXP iMX8m ARM
Memory: 4GB
Storage: 16GB
Security: TPM 2.0 Nuvoton NCPT750
Connectivity: WiFi & Bluetooth
Ports: 1xEthernet
1x USB C
2x USB A 3.0

Azure Percept
Trust Module

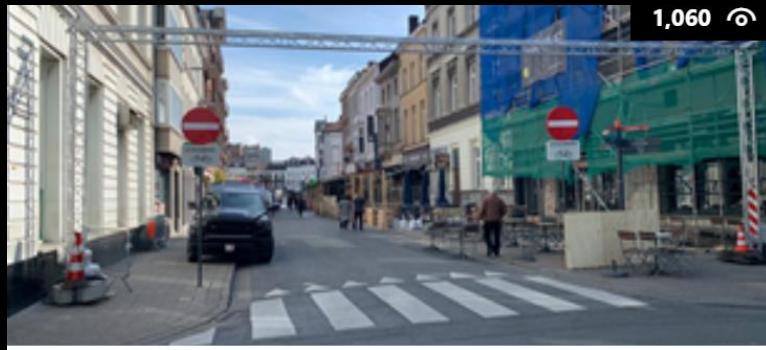


VPU: Intel Movidius Myriad X (MA2085)
Memory: LPDDR4 2GB
Sensor: Sony IMX-219 (1.12 um) RGB
Security: STM32L462CE
Resolution: 8MP @ 30FPS
Ports: USB C & MIPI 4 Lane (up to 1.5 Gbps per lane)
Ports: USB Micro B

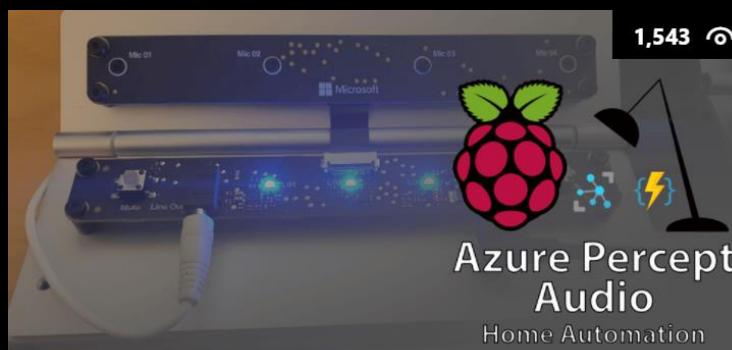
Azure Percept
Vision



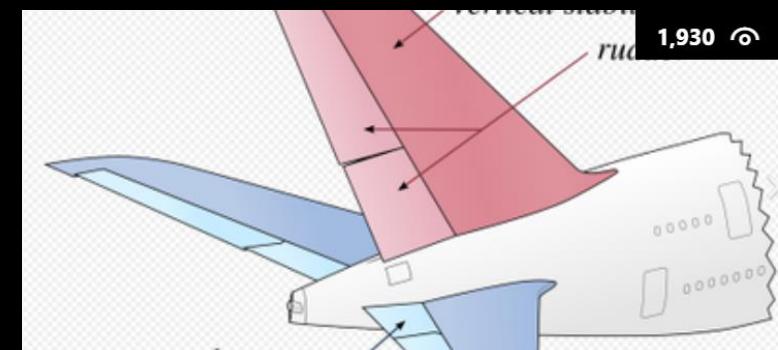
Real-World Scenarios...



[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 loss¹
- 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.

¹ 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



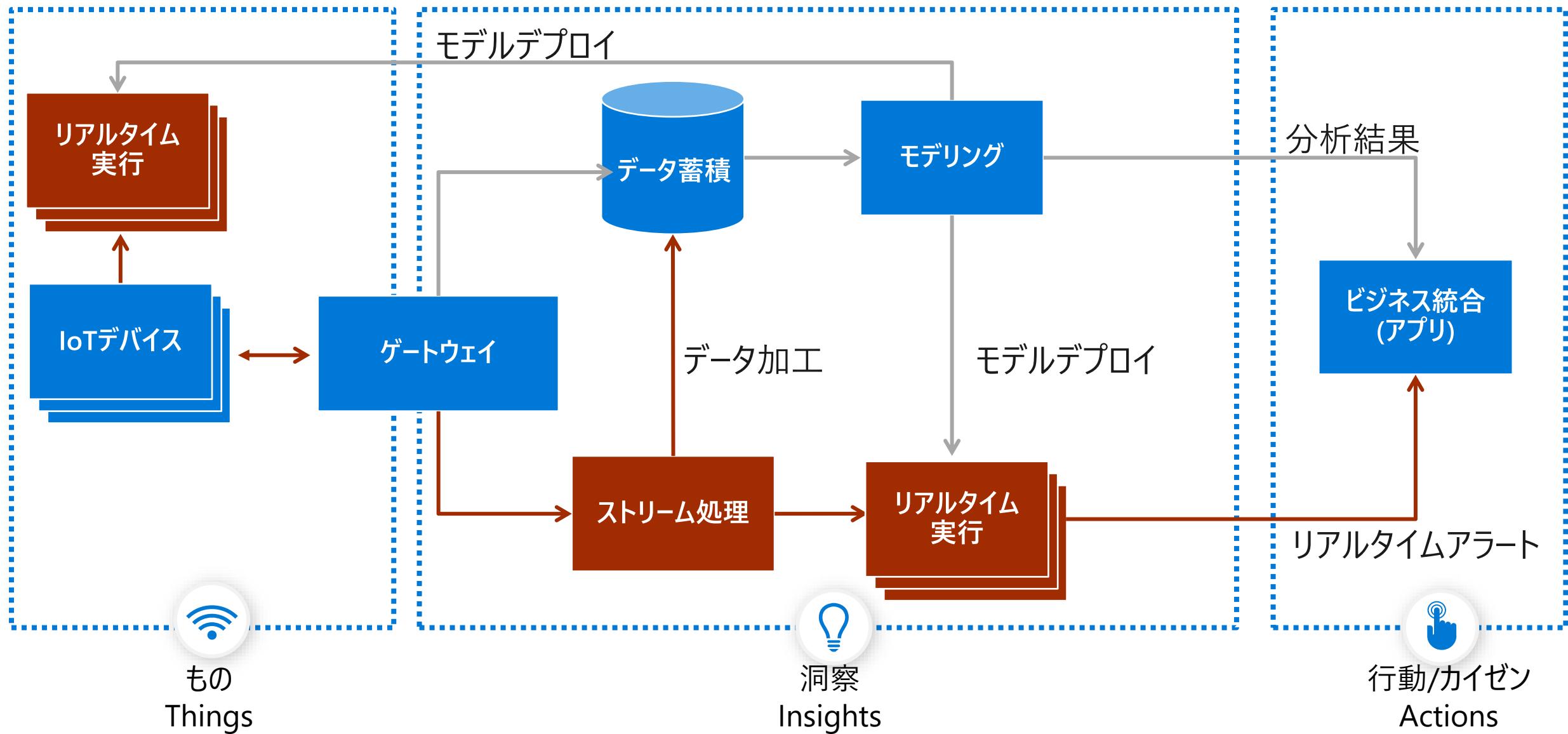
Percept AI output with Chickadee (*Poecile gambeli*) detection



Azure Percept ワークショップ

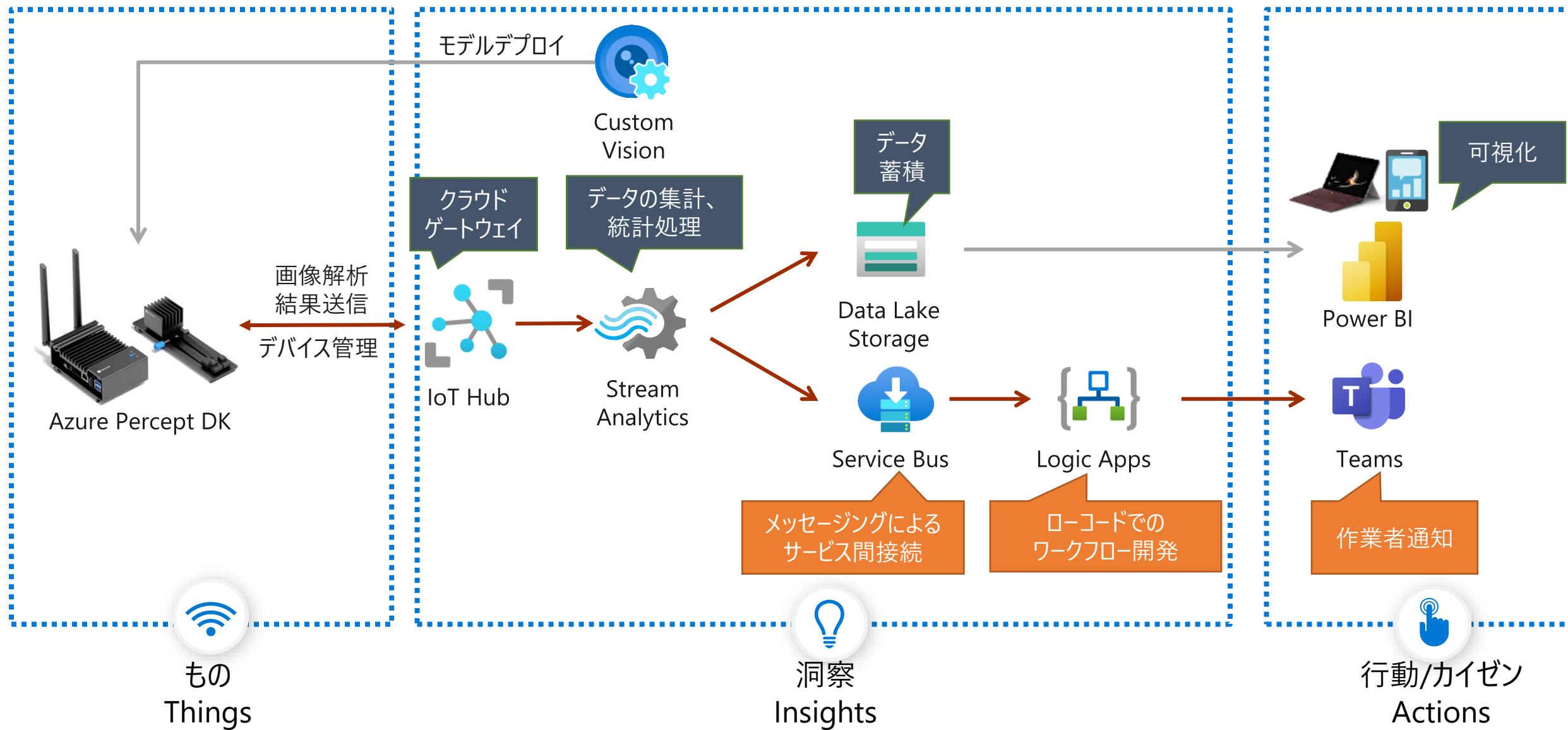
AI + IoTリファレンスアーキテクチャー

→ バッチ
→ リアルタイム



ハンズオン構成

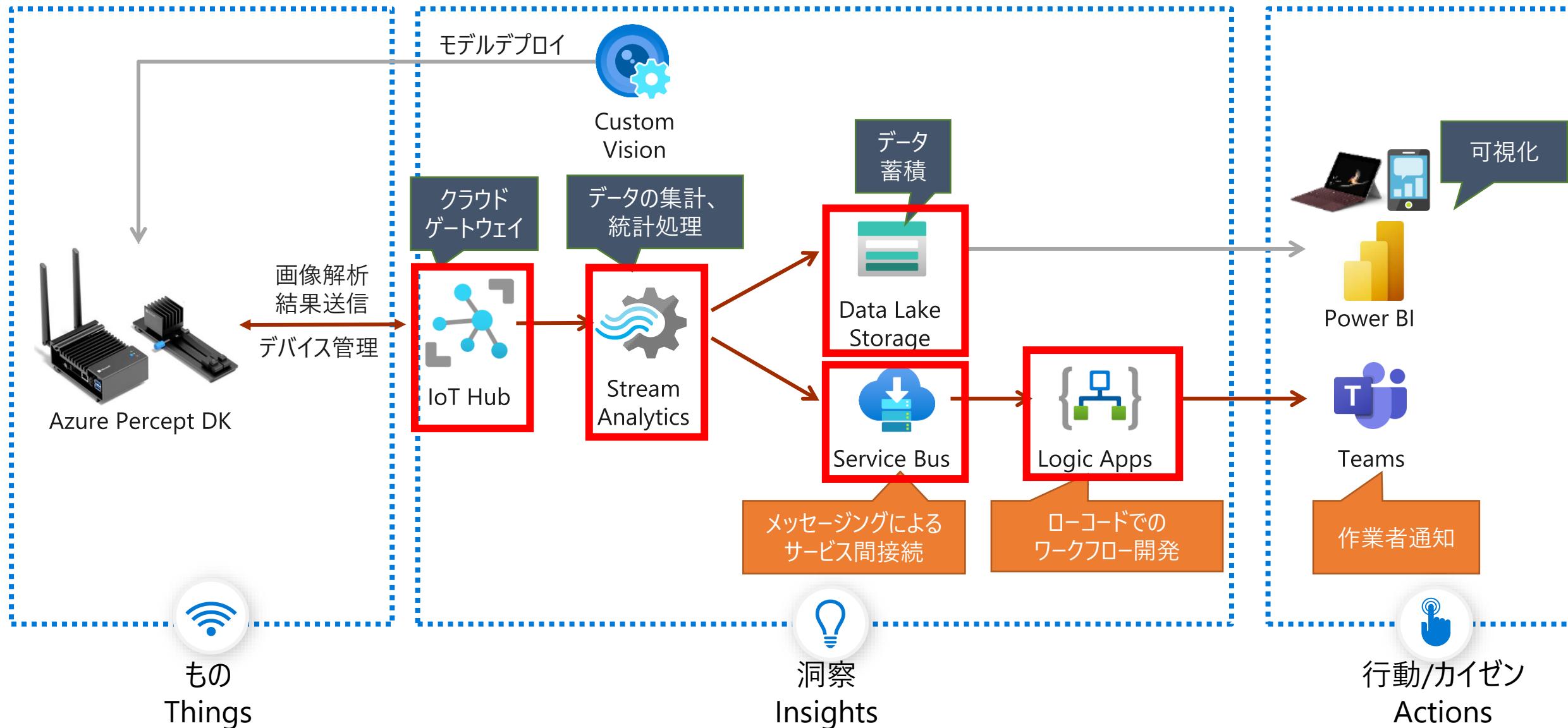
→ バッチ
→ リアルタイム



Step1: Azure環境のセットアップ

下記の赤枠の項目をテンプレートからAzure上にデプロイしていきます。

→ バッチ
→ リアルタイム



Step1: Azure環境のセットアップ

ワークショップ内で必要なAzure上のリソースのテンプレートをGitHubにご用意しました。

下記手順にてご利用ください。

「[GitHub - yahanda/hands-on-percept](#)」からGitHubにアクセスし、ページ下部にスクロールします。

「Simple version (without Synapse Analytics)」の下にある[Deploy to Azure]をクリックし、Azureの画面に遷移します。

(Microsoftアカウントへのサインイン画面が表示された場合、リソースのデプロイ先となるMicrosoftアカウントを選択し認証します。)

Deploy

Full version



Deploy to Azure

Japan West region is not available as of 2021 Dec.

Simple version (without Synapse Analytics)



Deploy to Azure

Step1: Azure環境のセットアップ

カスタムデプロイを実行します。

- ①今回のワークショップで使用するサブスクリプションを選択します。
- ②今回のワークショップで使用するリソースグループを選択します。
リソースグループ未作成の場合は[新規作成]から作成してください。
- ③デプロイ先となるリージョンを任意で選択します。
- ④Prefixとして任意の文字列を設定します。（デプロイされるリソースの先頭に付与される文字列となります。）
- ⑤[確認と作成]をクリックします。設定内容の検証に成功後に表示される[作成]をクリックするとデプロイが開始されます。
- ⑥デプロイ完了後、[リソースグループに移動]をクリックします。

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 * ①

MWC_Percept

リソース グループ * ①

(新規) MWC_Percept_RG

新規作成

インスタンスの詳細

リージョン * ①

西日本

Prefix ①

kenshou



✓ デプロイが完了しました

デプロイ名: Microsoft.Template-20211222163506
サブスクリプション: MWC_Percept
リソース グループ: MWC_Percept_RG

▽ 展開の詳細 (ダウンロード)

△ 次の手順

⑥ リソース グループに移動

⑤

確認と作成

< 前へ

次: 確認と作成 >

Step1: Azure環境のセットアップ

IoT Hub、ロジックアプリ、Service bus 名前空間、ストレージアカウントがそれぞれ存在することを確認できたらリソースのAzure環境のセットアップは完了です。

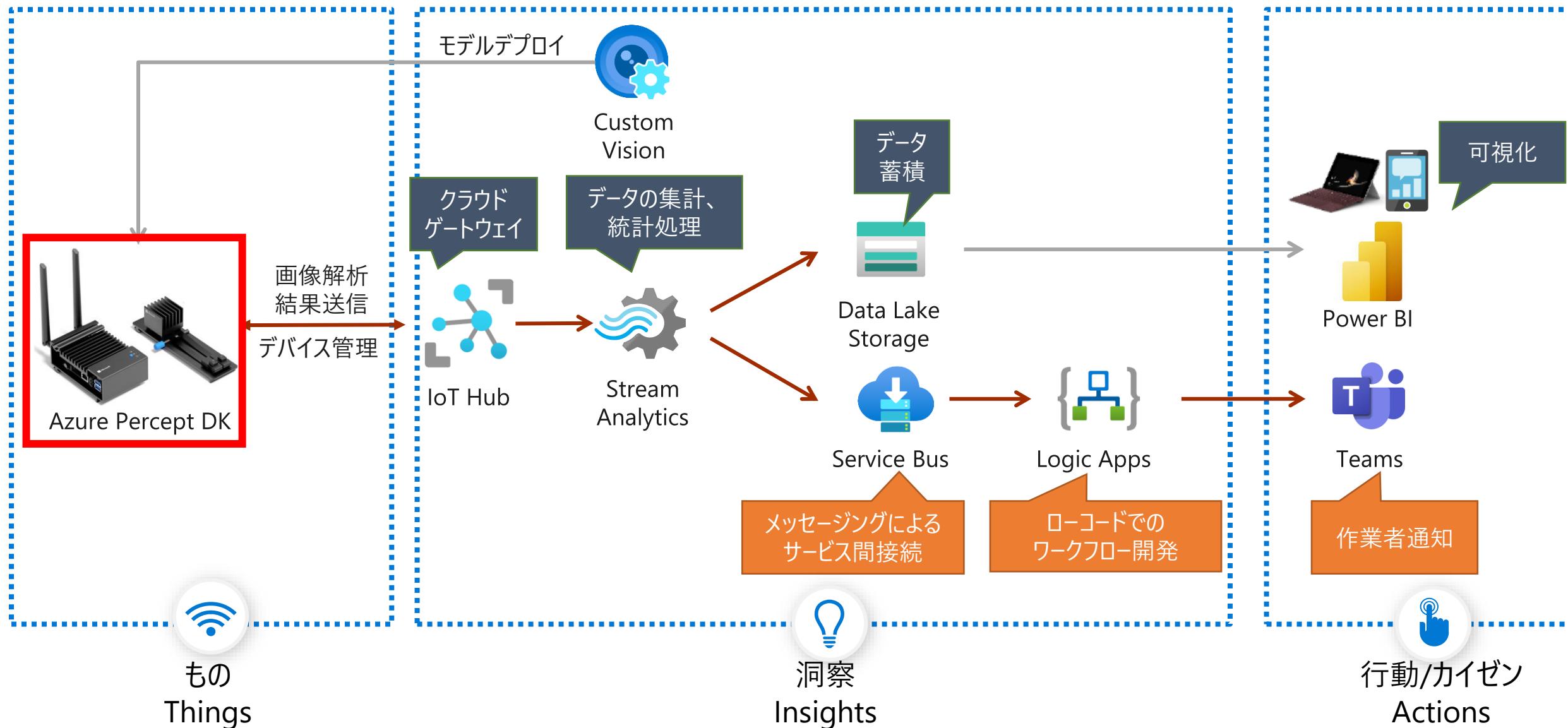
The screenshot shows the Azure portal's Resource Group settings page for 'MWC_Percept_RG'. The left sidebar lists navigation options: Overview, Activity Log, IAM, Tags, Resource Visualizer, Events, Settings, Deployment, Security, Policies, Properties, Locks, and Cost Management. The main content area displays basic information about the resource group, including its deployment status ('1 successful') and location ('West Japan'). It also shows a list of resources deployed under the group, which are highlighted with a red dashed border:

名前	種類	場所
kenshou-iohub-36rorl	IoT Hub	西日本
kenshou-logic-36rorl	ロジック アプリ	西日本
kenshou-servicebus-36rorl	Service Bus 名前空間	西日本
kenshoustorage36rorl	ストレージ アカウント	西日本

Step2: Percept DK のセットアップ

下記の赤枠の項目をセットアップしていきます。

→ バッチ
→ リアルタイム



Step2: Percept DK のセットアップ（組み立て）

Azure Percept DK コンポーネントを箱から取り出し組み立てます。

電源ケーブルをコンセントに接続すると、デバイスの電源が自動的にオンになるためコンセントの接続は最後に行います。

Before



After



Step2: Percept DK のセットアップ（組み立て）

- ①電源ケーブルをコンセントに接続後、Percept DKの左側にある電源ボタンが点灯し起動し始めます。
デバイスが起動するまでしばらく時間がかかります。
- ②「Wi-Fi」のLEDランプがゆっくり点滅し始めたら、Percept DKのネットワーク設定が可能になります。



①



②

Step2: Percept DK のセットアップ（設定）

ネットワーク設定とIoT-Hubへのデバイス登録を行います。

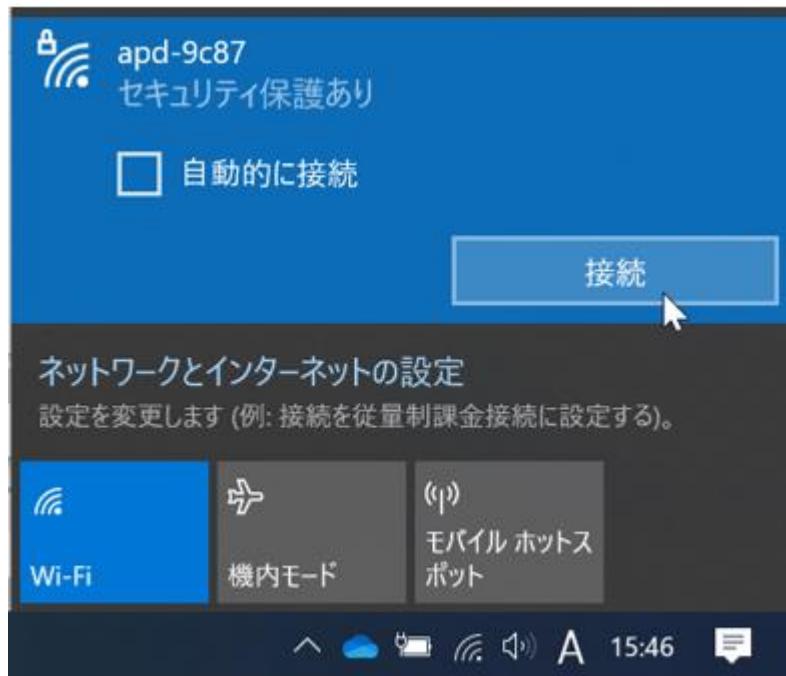
① PCの右下のWiFiアイコンをクリックし、ネットワークの一覧からPerceptDKのSSIDを探して接続してください。

② SSID及びそのパスワードは個体ごとに異なります。

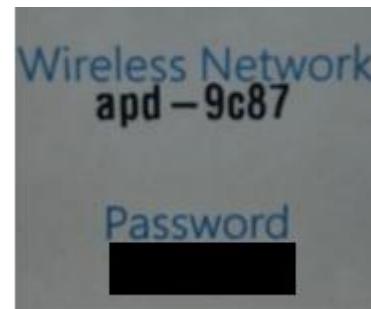
PerceptDKに同梱されていたWelcomeカードに記載されている情報をご参照ください。

③ 接続に成功すると自動でブラウザが起動します。[Next]をクリックして設定を開始します。

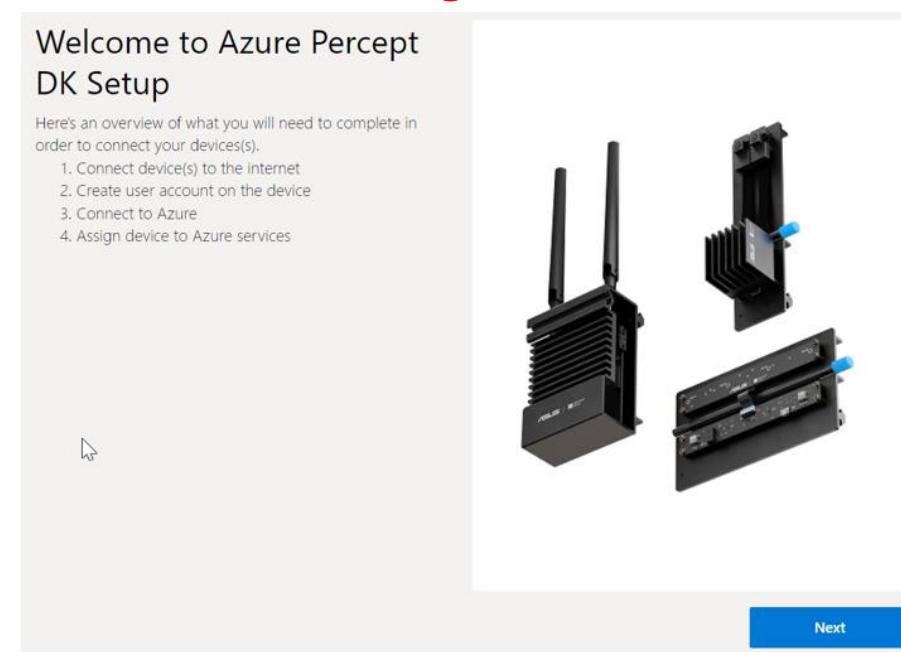
①



②



③



Step2: Percept DK のセットアップ（設定）

① [Connect to a new WiFi network]をクリックします。

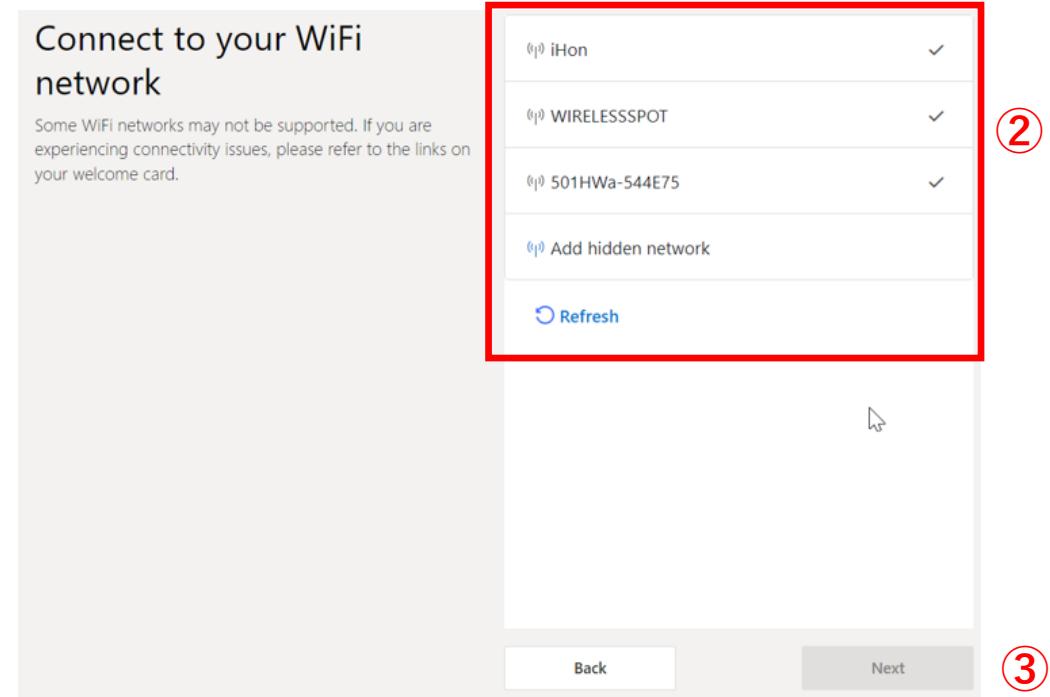
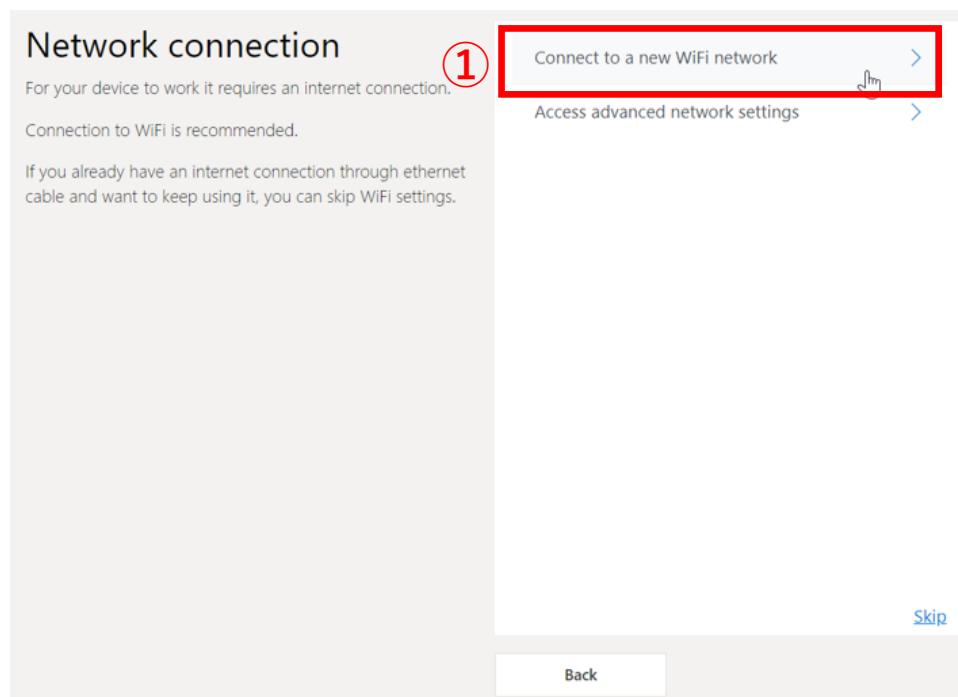
② 接続可能なネットワークの一覧が表示されます。

（これには少し時間がかかることがあります。あまりに遅い場合は本体のWi-Fiランプ横にある青いボタンを押し、

PerceptDKのネットワークを再起動して同様の操作を試してください。）

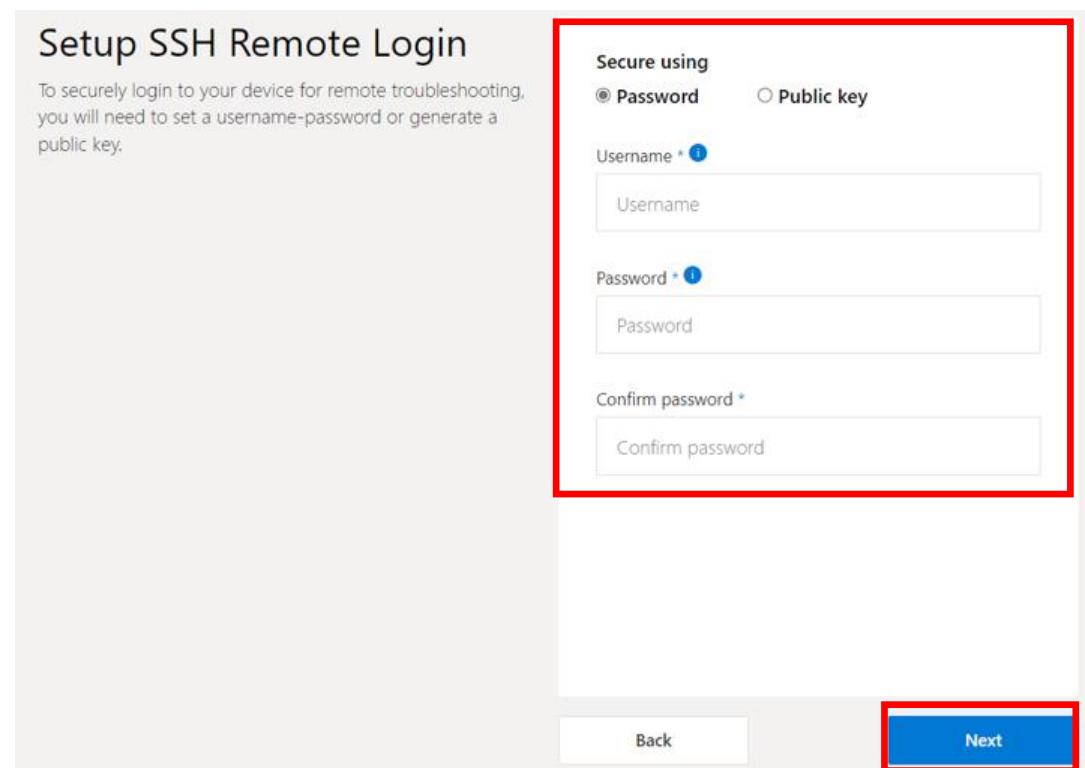
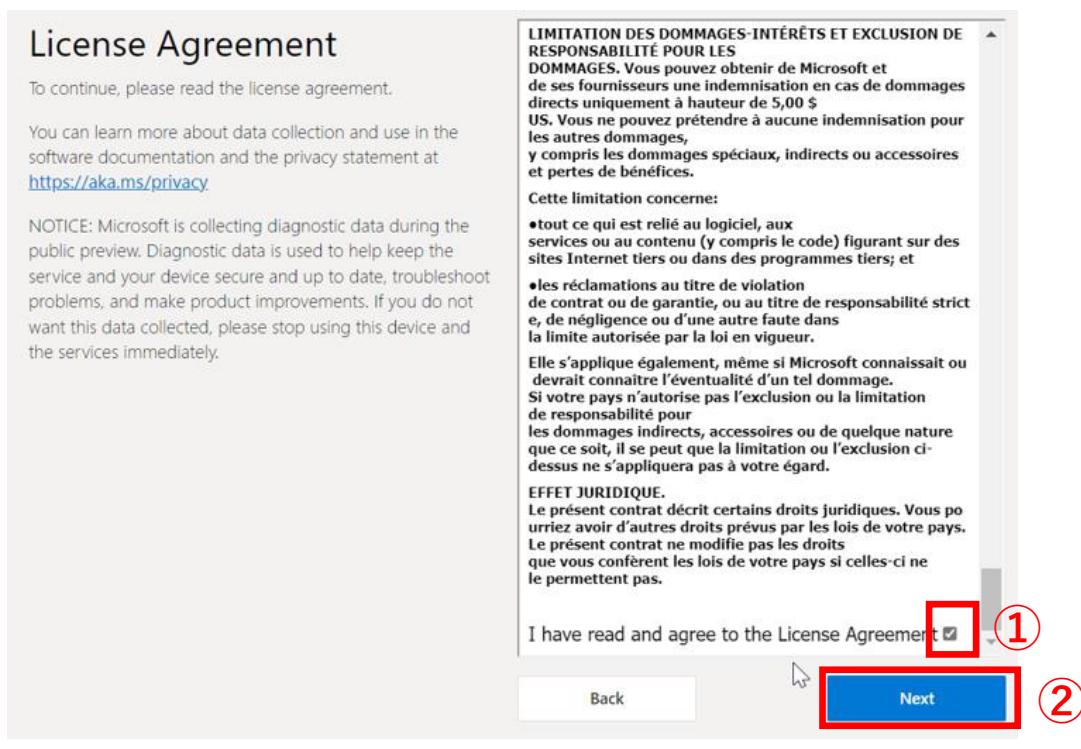
利用するネットワークを選択して接続します。多くの場合、パスワードによる認証が必要です。ネットワークの管理者に確認してください。

③ 接続に成功したら[Next]をクリックします。



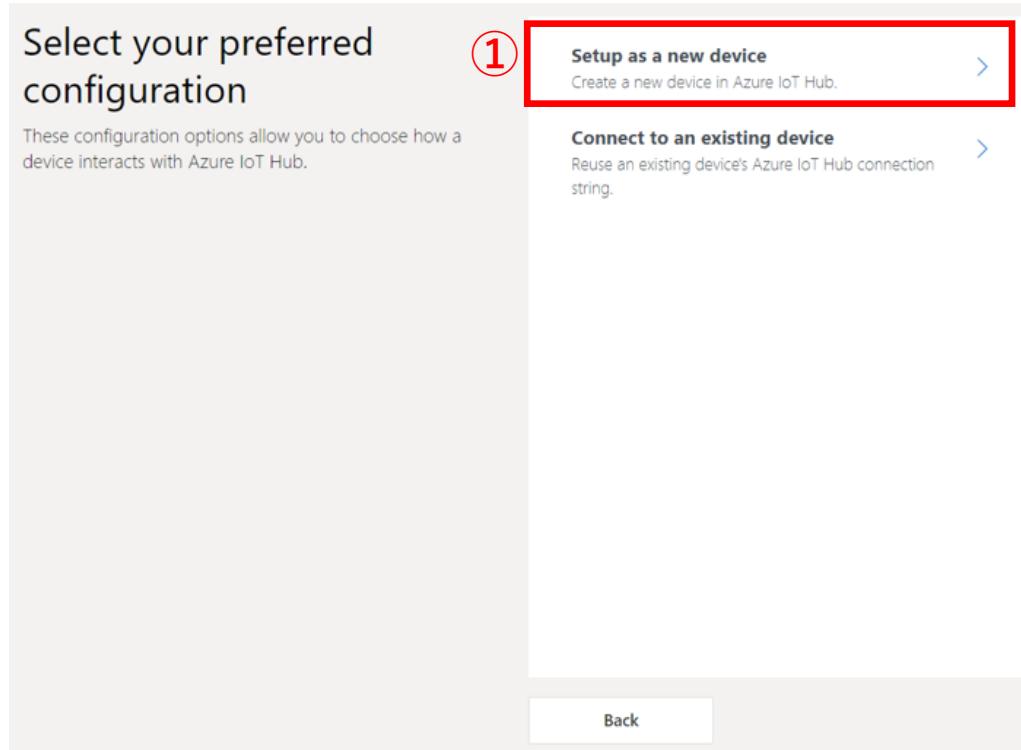
Step2: Percept DK のセットアップ（設定）

- ① License Agreementの内容を読みながら下へスクロールします。
「I have read and agree to the License Agreement」右横のチェックボックスにチェックを入れます。
- ② [Next]をクリックします。
- ③ SSH接続時に使用するPerceptDK上のユーザを作成します。
ユーザ名とパスワードを入力してください。今後使用するため、書き留めておいてください。
- ④ [Next]をクリックします。



Step2: Percept DK のセットアップ（設定）

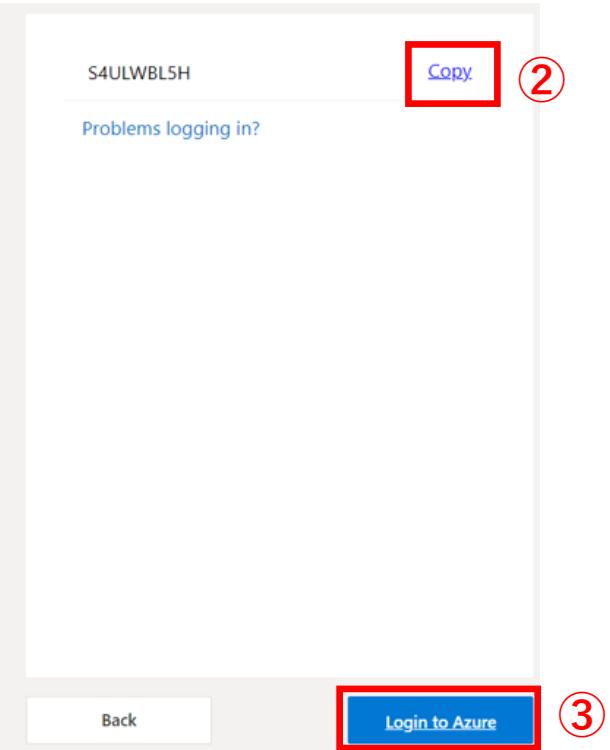
- ① [Setup as a new device]をクリックします。
- ② 自動で発行された一時的なパスワードを[copy]をクリックしてコピーします。
- ③ [Login to Azure]をクリックします。



Complete the authorization process

This code allows the device to create Azure resources on behalf of the user.

Copy this code and click the link below to complete the authorization process.



Step2: Percept DK のセットアップ（設定）

- ① 新たにコードの入力用のウィンドウが起動します。コピーしたパスワードを貼り付け、[次へ]をクリックします。
- ② サインインするか問われた場合は[続行]をクリックします。
- ③ このメッセージが表示されたらこのウィンドウ（タブ）を閉じます。

※次の手順からは前頁の画面から続きで操作するためブラウザは閉じないでください。



Step2: Percept DK のセットアップ（設定）

① Percept DKを登録するIoT-Hubを選択します。

「(任意に決めたPrefix)-iothub-(XXXXXX)」となっているIoT-Hubを選択します。

② IoT-Hub上のPerceptDKのデバイス名を任意で決めて入力します。

③ [Next]をクリックします。

Assign your device to your Azure IoT Hub

Select the Azure IoT Hub you would like to onboard your device to. If none are listed, select "Create a new Azure IoT Hub".

This Azure IoT Hub is linked to a Device Provisioning Service instance; allowing you to enroll using the TPM. 

[Learn more about Azure IoT Hub](#)

[Learn more about Azure Pricing](#)

kenshou-iothub-36rorl
Tier: Standard

①

Create a new Azure IoT Hub

>

Refresh

Name your Azure IoT Edge device

Give your device a unique name to identify it within kenshou-iothub-36rorl.

[Learn More About IoT Hub](#)

Azure IoT Edge Device name *

percept

②

Back

Next

③

Step2: Percept DK のセットアップ（設定）

このメッセージが表示されたらデバイスのセットアップが完了です。

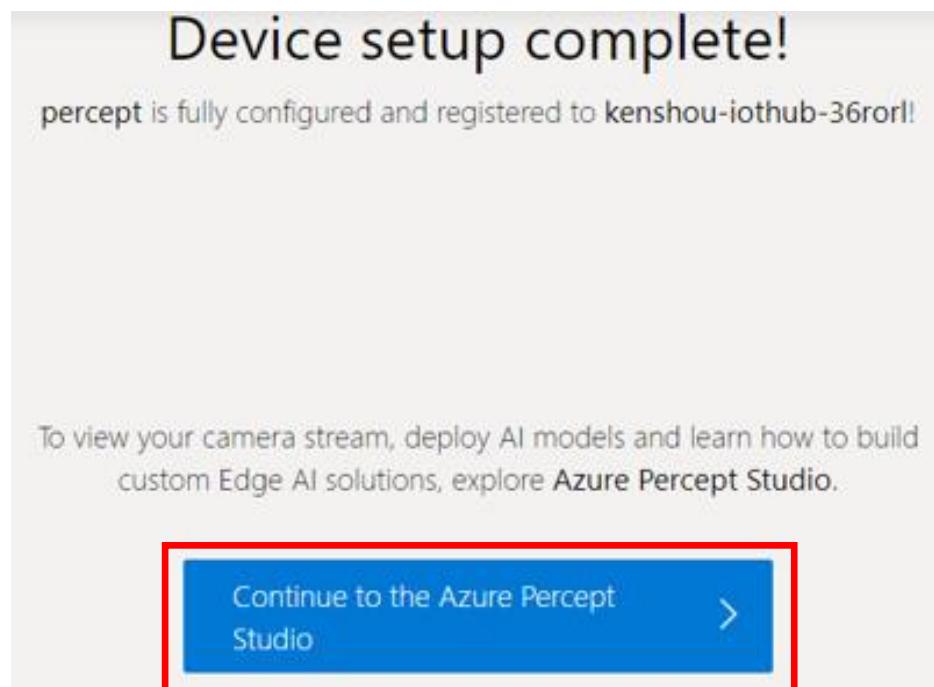
次にPCの接続先ネットワークを変更します。

この手順の時点ではPCはPerceptDKの展開するWi-Fiに接続しています。

タスクバーのネットワークアイコン（）等から、

PerceptDKの接続先としたものと同様のネットワークを選択し接続してください。

PCの接続先ネットワークを切り替えたら[Continue to the Azure Percept Studio]をクリックします。



Step2: Percept DK のセットアップ（設定）

- ① [デバイス]をクリックします。
- ② 任意の名前を設定したPerceptが表示され状態が「Connected」となっていることを確認します。
以上でPercept DKの設定は完了です。
- ③ 次に[(任意で設定したデバイス名)]をクリックします。

The screenshot shows the Azure Percept Studio interface for managing devices. The top navigation bar includes the Microsoft Azure logo, a search bar, and various icons. The main title is "Azure Percept Studio | デバイス". A sidebar on the left lists "AI プロジェクト" (1) with options for "Vision" and "音声" (Speech). Below this is a "サポートとトラブルシューティング" section with links for "ドキュメントとコミュニティ" and "新しいサポートリクエスト". The main content area displays a table of devices. The first row, which has a red border around its entire row, represents the device being managed. It contains the following information:

AI プロジェクト	デバイス名	状態	IoT ハブ	リソース グループ
① Vision	③ percept	② Connected	kenshou-iothub-36rorl	mwc_percept_rg

The columns are labeled: "AI プロジェクト" (AI Project), "デバイス名" (Device Name), "状態" (Status), "IoT ハブ" (IoT Hub), and "リソース グループ" (Resource Group). The device name "percept" and status "Connected" are highlighted with red boxes and numbered ③ and ② respectively. The row number ① is also indicated on the sidebar.

Step2: Percept DK のセットアップ（サンプルを試す）

サンプルのビジョン AI モデルを試します。

- ① [Vision]をクリックします。
- ② [サンプルモデルをデプロイします]をクリックします。

The screenshot shows the Microsoft Azure Percept Studio interface. At the top, there's a navigation bar with 'Microsoft Azure' and a search bar. Below it, the path 'ホーム > Azure Percept Studio >' is shown. The main area features a logo for 'percept' and 'Azure Percept DK'. Below the logo, there are three tabs: '全般' (General), 'Vision' (which is highlighted with a red box and has a red circled '1' above it), and '音声' (Voice). Underneath these tabs are three buttons: '最新の情報に更新' (Update latest information), 'デバイスを IoT Hub で開く' (Open device in IoT Hub), and 'トラブルシューティング' (Troubleshooting). The 'Vision' section contains a 'Vision モジュールのバージョン' (Vision module version) with a link to 'preload-devkit' and a note about SW update. The 'Actions' section at the bottom has four items: 'プロジェクトのイメージをキャプチャ' (Capture project image) with a link to 'イメージをキャプチャします', 'デバイス ストリームの表示' (Display device stream) with a link to 'ストリームを表示します', 'Custom Vision プロジェクトのデプロイ' (Deploy Custom Vision project) with a link to 'プロジェクトをデプロイします', and 'サンプル モデルのデプロイ' (Deploy sample model) with a link to 'モデルをデプロイします'. The 'モデルをデプロイします' button is highlighted with a red box and has a red circled '2' above it. A note at the bottom states: 'ストリームを表示するには、デバイスが同じネットワークにある必要があります' (To display the stream, the device must be on the same network).

Step2: Percept DK のセットアップ（サンプルを試す）

① [一般的なオブジェクトの検出]にチェックを入れます。

② [デバイスへのデプロイ]をクリックします。

モデルのデプロイが完了するとブラウザ右上に完了メッセージが表示されます。



Step2: Percept DK のセットアップ（サンプルを試す）

[ストリームを表示します]をクリックします。

新たにウィンドウ（タブ）が立ち上がり、Webstream Videoが表示されます。

Webstream Videoではリアルタイムに映像が写しだされ一般的なオブジェクトが認識されます。



ホーム > Azure Percept Studio >

percept ...
Azure Percept DK

全般 Vision 音声

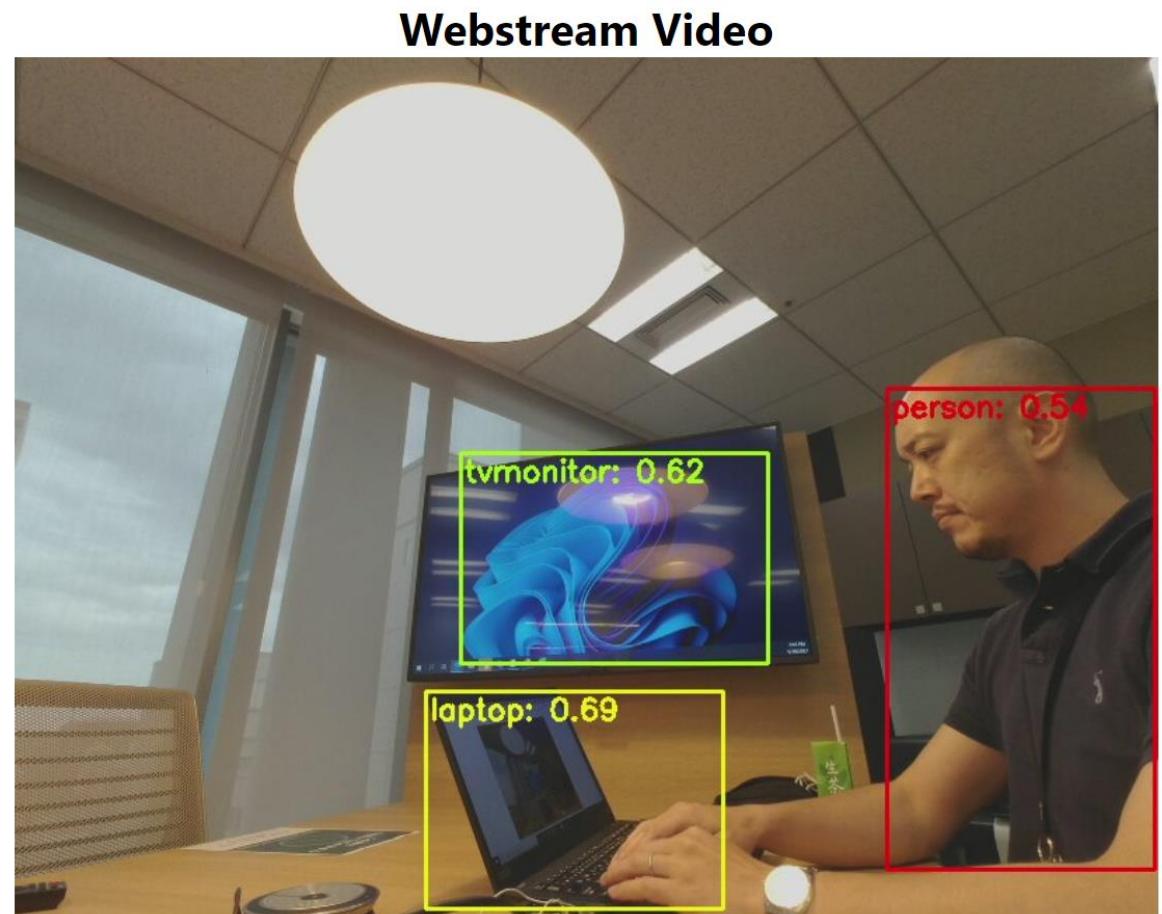
最新の情報に更新 デバイスを IoT Hub で開く トラブルシューティング

Vision モジュールのバージョン ①
preload-devkit ② モジュールのアップグレードには SW 更新プログラムが必要です

アクション

- プロジェクトのイメージをキャプチャ
イメージをキャプチャします
- デバイス ストリームの表示
ストリームを表示します

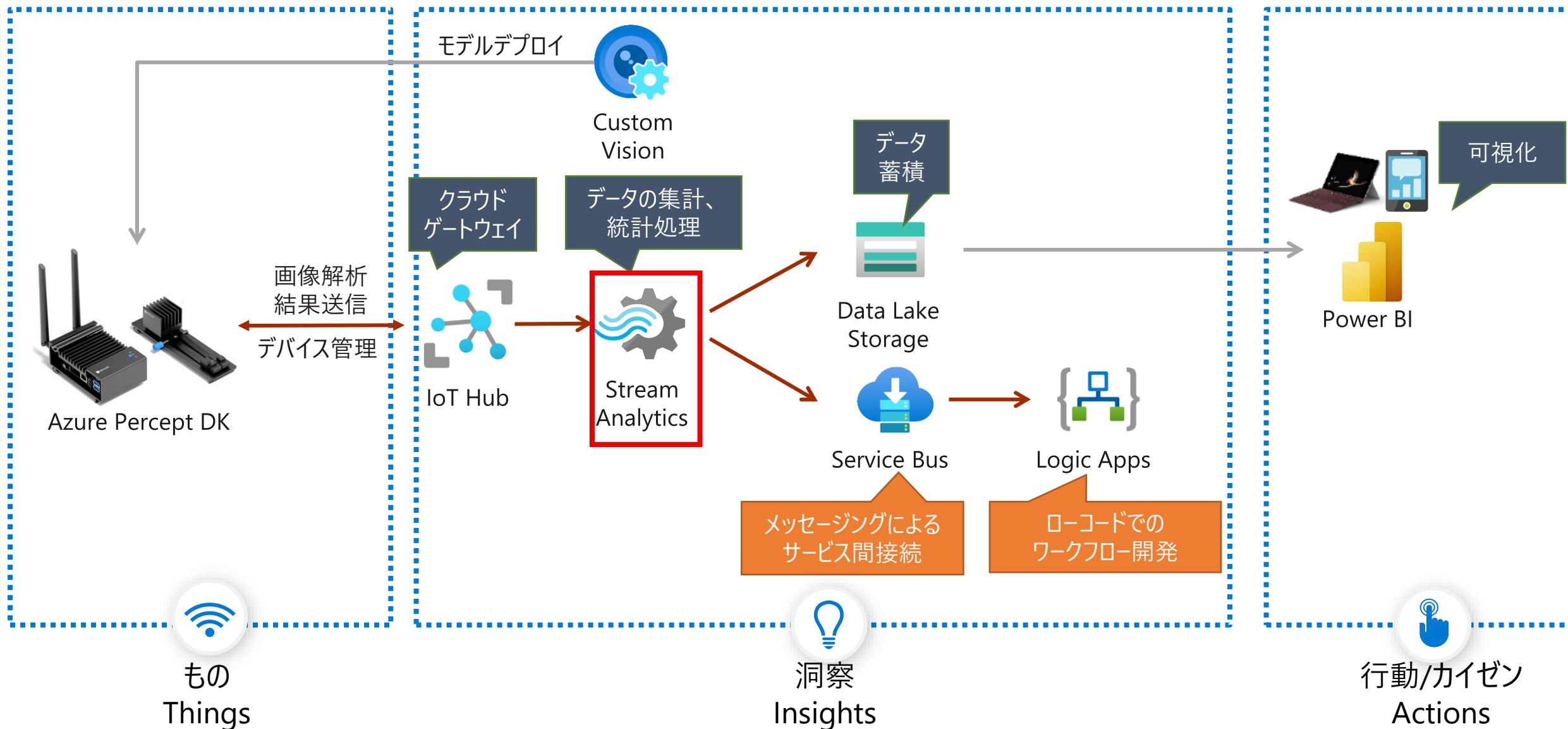
ストリームを表示するには、デバイスが同じネットワークにある必要があります



Step3: Stream Analyticsの起動

下記の赤枠の項目をセットアップしていきます。

→ バッチ
→ リアルタイム



Step3: Stream Analyticsの起動

Azure Portalに戻り、「Stream Analytics」で検索する等して「Stream Analytics ジョブ」の一覧を表示します。さらに今回デプロイしたストレージアカウントである[(任意に決めたPrefix)-asa(XXXXXX)]をクリックします。

The screenshot shows the Microsoft Azure portal interface. On the left, the navigation bar has 'Stream Analytics' selected. A large red box highlights the 'Stream Analytics' link in the search bar and the 'Stream Analytics ジョブ' link in the service list. A large grey arrow points from the left panel to the right panel. The right panel is titled 'Stream Analytics ジョブ' and shows a single item in the list: 'kenshou-asa-36rorl'. This item is also highlighted with a red box. The list includes columns for '名前' (Name) and 'リソース グループ' (Resource Group), with 'kenshou-asa-36rorl' under '名前' and 'MWC_Percept_RG' under 'リソース グループ'.

Step3: Stream Analyticsの起動

- ① 概要に表示されている[開始]をクリックすると、データの集計/統計処理が開始されます。
- ② 収集しているデータの内容は[クエリ]、[iothub]の順にクリックすると表示されます。



The screenshot shows the Stream Analytics query editor for the job 'kenshou-asa-36rorl'. The left sidebar lists '概要', 'アクティビティログ', 'タグ', '問題の診断と解決', '設定', 'プロパティ', 'ロック', 'ジョブトポロジ', '入力', '関数', 'クエリ' (highlighted with a red box and number 2), and '構成'. The main area shows the query code:

```
<> kenshou-asa-36rorl | クエリ ...
Stream Analytics ジョブ

検索 (Ctrl+ /) << クエリ言語のドキュメント Visual Studio で開く UserVoice

概要
アクティビティログ
アクセス制御 (IAM)
タグ
問題の診断と解決
設定
プロパティ
ロック
ジョブトポロジ
入力
関数
クエリ (highlighted)
構成
環境
ストレージ アカウントの設定

クエリのテスト クエリの保存 変更の破棄

<> クエリ (1)
</> iothub
出力 (2)
</> adls
</> servicebus

クエリのテ스트
SELECT
Percept.ArrayValue.label,
Percept.ArrayValue.confidence,
GetArrayElement(Percept.ArrayValue.bbox, 0) AS bbox0,
GetArrayElement(Percept.ArrayValue.bbox, 1) AS bbox1,
GetArrayElement(Percept.ArrayValue.bbox, 2) AS bbox2,
GetArrayElement(Percept.ArrayValue.bbox, 3) AS bbox3,
CAST (udf.main(Percept.ArrayValue.timestamp) as DateTime) as DETECTION_TIMESTAMP,
Percept.ArrayValue.timestamp
INTO
[adls]
FROM
[iothub] as event
CROSS APPLY GetArrayElements(event.Neural_Network) AS Percept
WHERE
CAST(Percept.ArrayValue.confidence as Float) > 0.5
SELECT
Percept.ArrayValue.label,
COUNT(*),
System.Timestamp AS WinEndTime
INTO
[servicebus]
FROM
[iothub] as event
CROSS APPLY GetArrayElements(event.Neural_Network) AS Percept
GROUP BY Percept.ArrayValue.label, HoppingWindow(second, 10, 5)
```

Step3: Stream Analyticsの起動 (補足)

```
1  SELECT
2    Percept.ArrayValue.label,
3    Percept.ArrayValue.confidence,
4    GetArrayElement(Percept.ArrayValue.bbox, 0) AS bbox0,
5    GetArrayElement(Percept.ArrayValue.bbox, 1) AS bbox1,
6    GetArrayElement(Percept.ArrayValue.bbox, 2) AS bbox2,
7    GetArrayElement(Percept.ArrayValue.bbox, 3) AS bbox3,
8    CAST (udf.main(Percept.ArrayValue.timestamp) as DateTime) as DETECTION_TIMESTAMP,
9    Percept.ArrayValue.timestamp
10   INTO
11     [adls]
12   FROM
13     [iothub] as event
14   CROSS APPLY GetArrayElements(event.Neural_Network) AS Percept
15   WHERE
16     CAST(Percept.ArrayValue.confidence as Float) > 0.5
17
18   SELECT
19     Percept.ArrayValue.label,
20     COUNT(*),
21     System.Timestamp AS WinEndTime
22   INTO
23     [servicebus]
24   FROM
25     [iothub] as event
26   CROSS APPLY GetArrayElements(event.Neural_Network) AS Percept
27   GROUP BY Percept.ArrayValue.label, HoppingWindow(second, 10, 5)
```

クエリの概要を説明します。

2行目:検出項目名

3行目:信頼度

4-7行目:検出時の枠の4つ角の座標

8行目:検出時刻

9行目:Perceptのシステム時刻

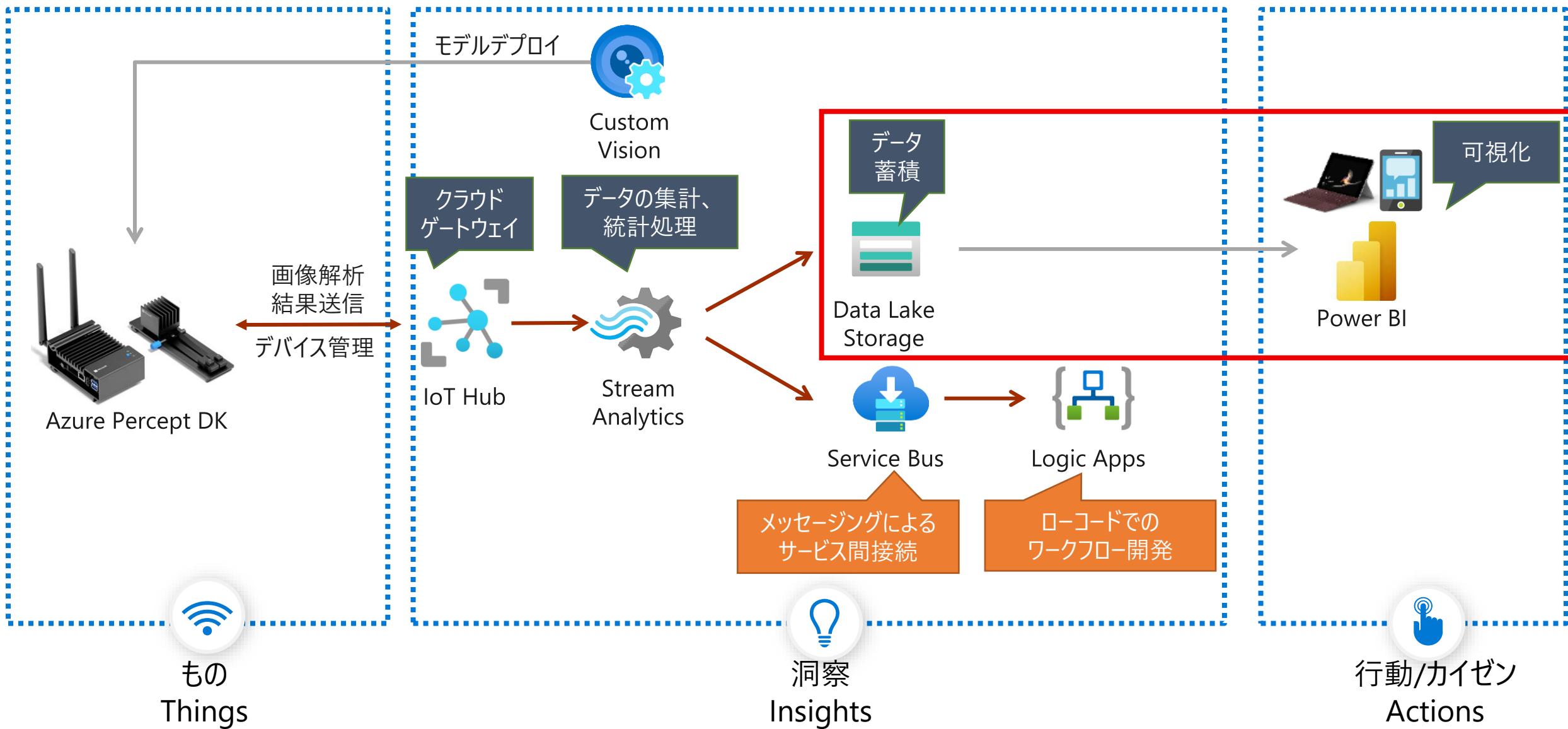
16行目:信頼度0.5以上で検出

27行目:ウィンドウ関数で5秒に1度、
直近10秒の情報を元に判断

Step4: Power BI レポートの作成

下記の赤枠の項目をセットアップしていきます。

→ バッチ
→ リアルタイム



Step4: Power BI レポートの作成

Azure Portalに戻り、「Storage Account」で検索する等してストレージアカウントの一覧を表示します。さらに今回デプロイしたストレージアカウントである[(任意に決めたPrefix)storage(XXXXX)]をクリックします。



The image shows two screenshots of the Microsoft Azure portal. The left screenshot shows the main Azure portal homepage with the 'Storage Account' service selected in the sidebar. The right screenshot shows the 'Storage Account' blade, displaying a list of storage accounts. One account, 'kenshoustorage36r0rl', is highlighted with a red box.

名前	種類	サブタイプ
kenshoustorage36r0rl	ストレージ アカウント	StorageV2

Step4: Power BI レポートの作成

- ① [アクセスキー]をクリックします。
- ② [キーの表示]をクリックします。
- ③ ストレージアカウントアカウント名をコピーしてテキストエディタに貼り付けるなどして控えます。
- ④ キーをコピーしてテキストエディタに貼り付けるなどして控えます。



Step4: Power BI レポートの作成

- ① Power BI を起動します。
- ② ホームタブの[データを取得]をクリックします。
- ③ [Azure]をクリックします。
- ④ [Azure Blob Storage]をクリックします。



①



データを取得

検索

すべて

ファイル

データベース

Power Platform

③ Azure

オンライン サービス

その他

Azure

Azure SQL Database

Azure Synapse Analytics SQL

Azure Analysis Services データベース

Azure Database for PostgreSQL

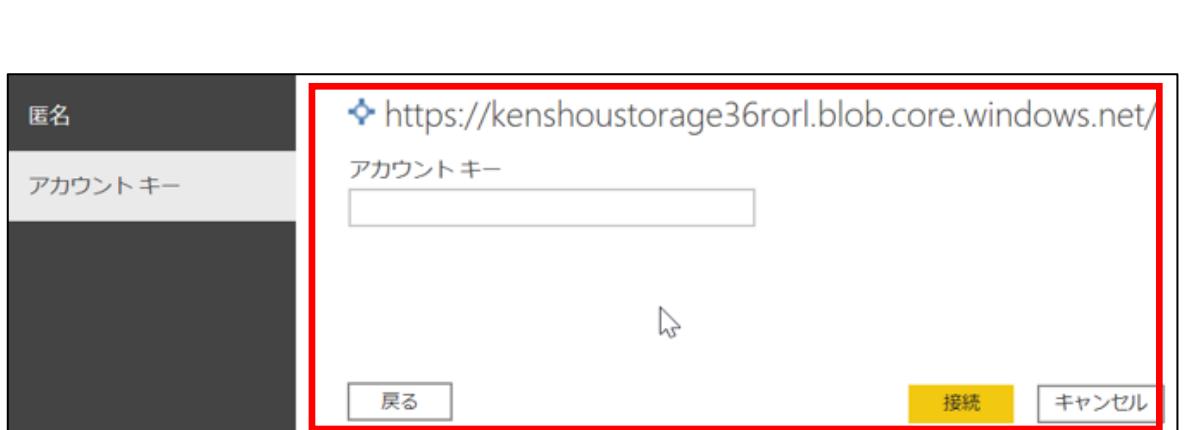
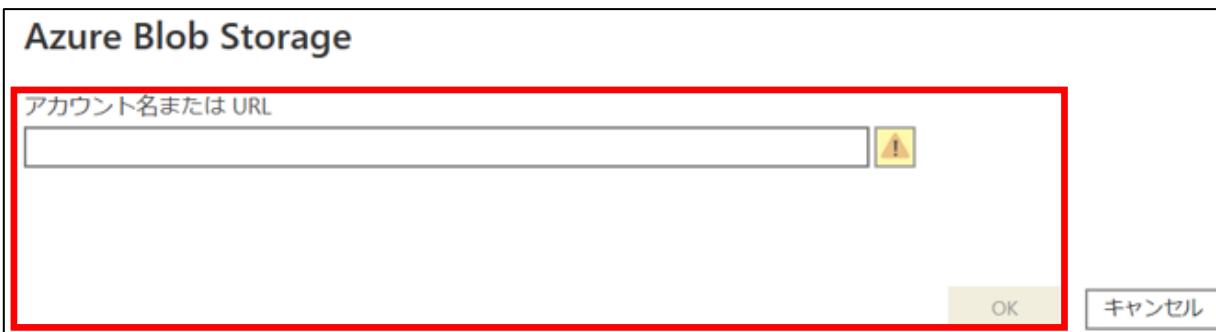
④ Azure Blob Storage

Azure Table Storage ④

Azure Cosmos DB

Step4: Power BI レポートの作成

- ① コピーしておいたストレージアカウント名を貼り付けます。
- ② コピーしておいたキーを貼り付けます。
- ③ [Azure]をクリックします。
- ④ [Azure Blob Storage]をクリックします。



Content	Name	Extension
Binary	stream	
Binary	stream/device	
Binary	stream/device/2021	
Binary	stream/device/2021/12	
Binary	stream/device/2021/12/22	
Binary	stream/device/2021/12/22/-1813886076_b499ff44241c4fa1ad11e2213c5.csv	

Step4: Power BI レポートの作成

① Extension列の値が「.csv」の行の[Binary] をクリックします。

② [閉じて適用]をクリックします。

今回カメラが検出している値を元データとしてレポートを作成できる状態になります。

	Content	Name
1	Binary	stream
2	Binary	stream/device
3	Binary	stream/device/2021
4	Binary	stream/device/2021/12
5	Binary	stream/device/2021/12/22
6	Binary	stream/device/2021/12/22/-1813886076_b499ff44241c4fa1ad11e22... .csv

無題 - Power Query エディター

ホーム

新しい クエリ

データ ソース

パラメーター

クエリ

列の管理

クエリ [1]

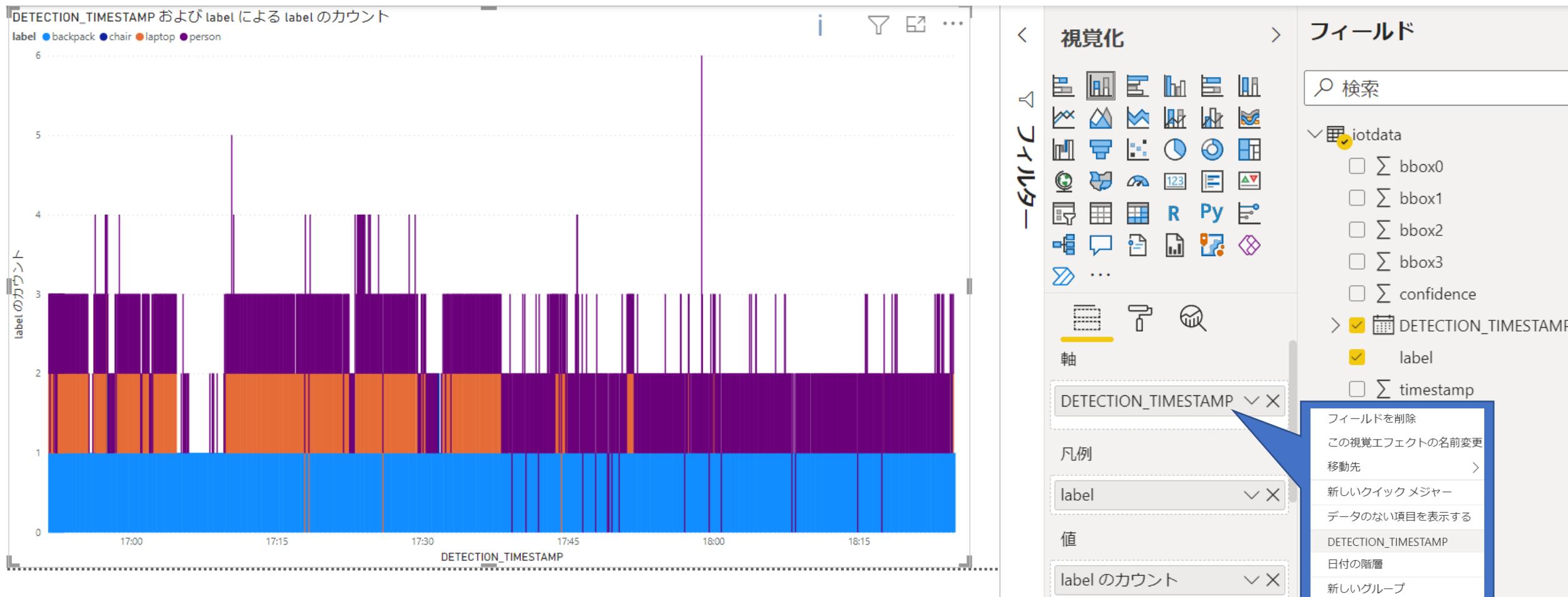
iotdata

	label	confidence	bbox0
1	person	0.809082	
2	backpack	0.602596	
3	laptop	0.537387	
4	person	0.780273	
5	backpack	0.609068	
6	laptop	0.561627	
7	person	0.817383	
8	backpack	0.580106	

Step4: Power BI レポートの作成

「視覚化」でグラフを選択し、「フィールド」から必要なデータを軸・凡例・値などにドラッグアンドドロップで割り当ててレポートを作成できます。検出したデータをPowerBIで見える化する手順の説明は以上となります。

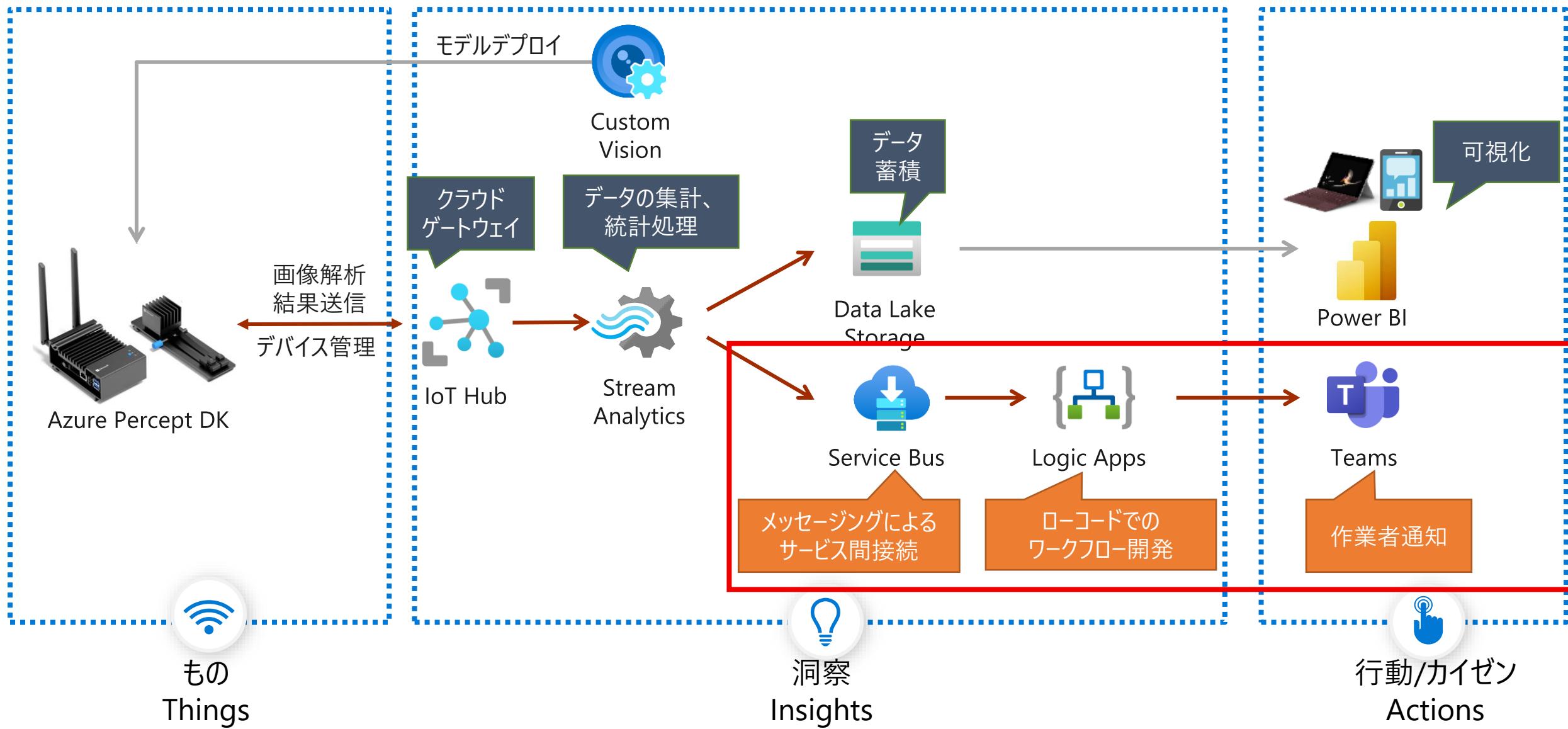
(下図の例では縦棒グラフを選択し、「label」→「値」、「label」→「凡例」、「DETECTION_TIMESTAMP」→「軸」としてさらに軸としたDETECTION_TIMESTAMPを右クリックして[DETECTION_TIMESTAMP]を選択しています。)



Step5: Teams アラートの実装

下記の赤枠の項目をセットアップしていきます。

→ バッチ
→ リアルタイム



Step5: Teams アラートの実装（Service Bus）

Azure Portalに戻り、「Service Bus」で検索する等してService Busの一覧を表示します。

さらに今回デプロイしたストレージアカウントである[(任意に決めたPrefix)-servicebus-(XXXXXX)]をクリックします。

The screenshot shows the Microsoft Azure portal interface. On the left, there is a search bar with the placeholder 'Service Bus' and a red box highlighting it. Below the search bar, under the 'サービス' (Services) section, there is a 'Service Bus' item with a red box highlighting it. To the right, a large grey arrow points from the search results towards a detailed view of the Service Bus service. The detailed view shows a header with 'Service Bus' and various actions like '作成' (Create), '更新' (Update), and 'CSV'. Below the header, there is a search bar with '任意のフィールドのフィルター...' (Filter by any field) and a 'サブスクリプション' (Subscription) dropdown. A message indicates '1 件中 1 ~ 1 件のレコードを表示しています.' (1 record displayed). At the bottom, there is a table with one row, which is also highlighted with a red box. The table row contains a checkbox, a small blue icon, and the text 'kenshou-servicebus-36rorl'.

Step5: Teams アラートの実装（Service Bus）

エンティティの「キュー」をクリックします。

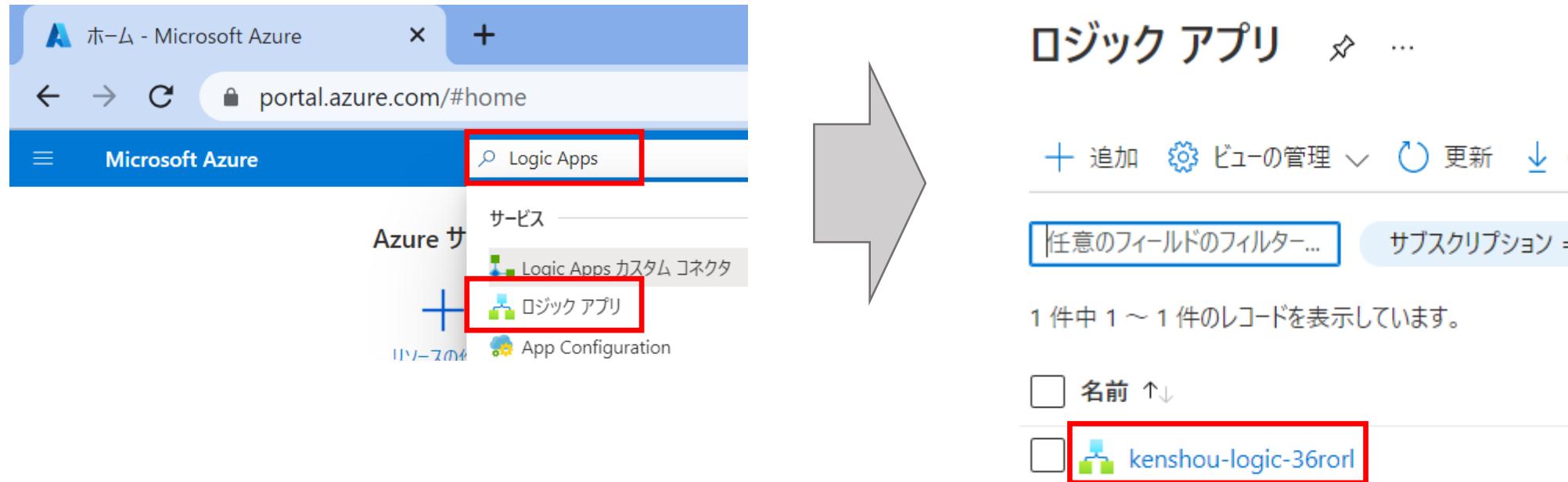
キュー「alertqueue」がActiveとなっていることを確認します。

The screenshot shows the Azure Service Bus Queue list page for the namespace 'kenshou-servicebus-36rorl'. The left sidebar has buttons for 'エンティティ' (Entity), 'キュー' (Queue) which is highlighted with a red box, and 'トピック' (Topic). The main area has a search bar, a 'キュー' (Queue) button, and a '最新の情報に更新' (Update latest information) button. A filter bar says '検索して項目をフィルター処理...' (Filter items by search). The table lists one queue:

名前	状態	最大サイズ
alertqueue	Active	1024 MB

Step5: Teams アラートの実装 (Logic Apps)

Azure Portalに戻り、「Logic Apps」で検索する等してLogic Appsの一覧を表示します。
さらに今回デプロイしたストレージアカウントである[(任意に決めたPrefix)-logic-(XXXXXX)]をクリックします。



Step5: Teams アラートの実装 (Logic Apps)

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]を選択します。
- 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] を選択します。



Step5: Teams アラートの実装 (Logic Apps)

- ① [すべて] タブを選択して、[Service Bus]を選択します。
- 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)]を選択します。

The image shows the Microsoft Logic Apps interface. On the left, a search bar and a navigation bar with tabs like 'おすすめ' and 'すべて' are visible. Below the tabs, a grid of icons for various connectors. The 'Service Bus' icon is highlighted with a red box and a red number '1'. A large grey arrow points from this screen to the right. On the right, a list of triggers under the heading 'トリガー' is shown. One specific trigger, '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)' (Queue trigger - When a message is received in a queue (Auto-complete)) is highlighted with a red box and a red number '2'. The full list of triggers is as follows:

- 1つ以上のメッセージがキューに届いたとき (ピーク ロック)
Service Bus
- 1つ以上のメッセージがトピックに届いたとき (ピーク ロック)
Service Bus
- 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)
Service Bus
- トピックで 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)
Service Bus
- メッセージがキューに着信したとき (オート コンプリート)
Service Bus
- メッセージがキューに着信したとき (ピーク ロック)
Service Bus
- メッセージがトピック サブスクリプションに着信したとき (オート コンプリート)
Service Bus
- メッセージがトピック サブスクリプションに着信したとき (ピーク ロック)
Service Bus

At the bottom, there is a message: '必要な情報が表示されませんか?' and a note: '次に追加するコネクタやトリガーについて、要望をお寄せください UserVoice'.

Step5: Teams アラートの実装 (Logic Apps)

- 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します。
- ② [項目を確認する頻度] を3秒に設定します。
- ③ [+新しいステップ] を選択します。



Step5: Teams アラートの実装 (Logic Apps)

- 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]を選択します。
- ② [アクション]で、[条件]を選択します。



Step5: Teams アラートの実装 (Logic Apps)

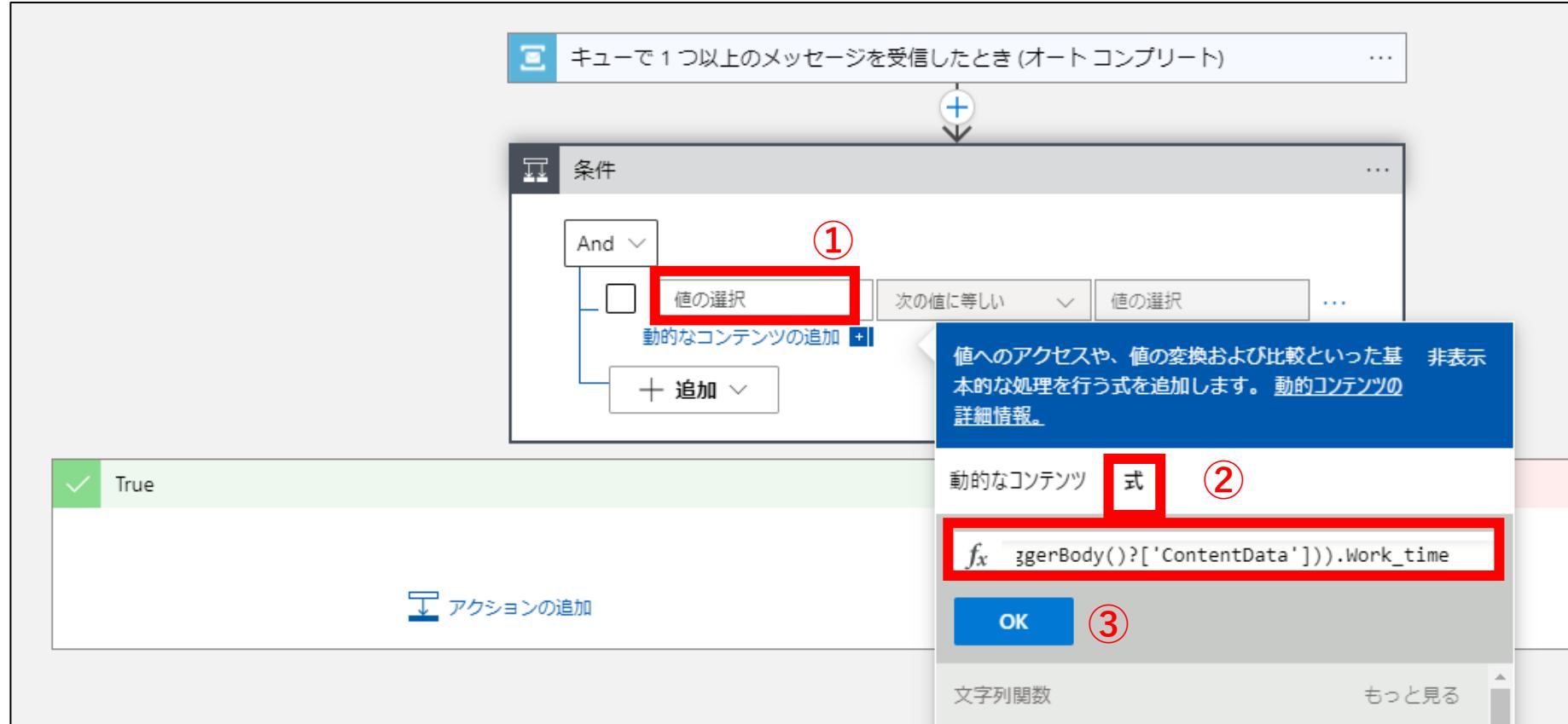
① [条件] にて [値の選択] を選択します。

② [式] タブを選択して、以下式を入力します。

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

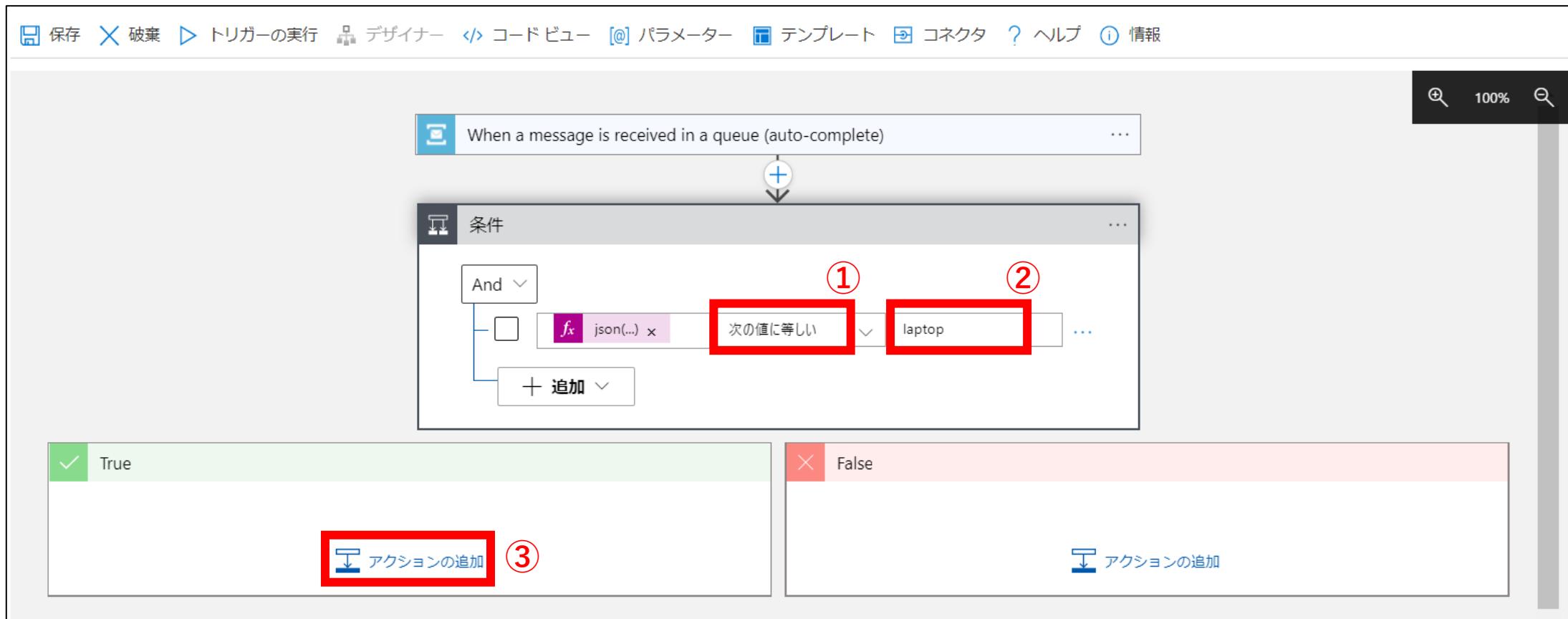
Stream Analyticsで出力した label を取得

③ [OK] を選択します。



Step5: Teams アラートの実装 (Logic Apps)

- ① ドロップダウンより、[次の値に等しい] を選択します。
- ② 該当ラベル名(例: laptop) を入力欄に直接入力します。※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] を選択します。



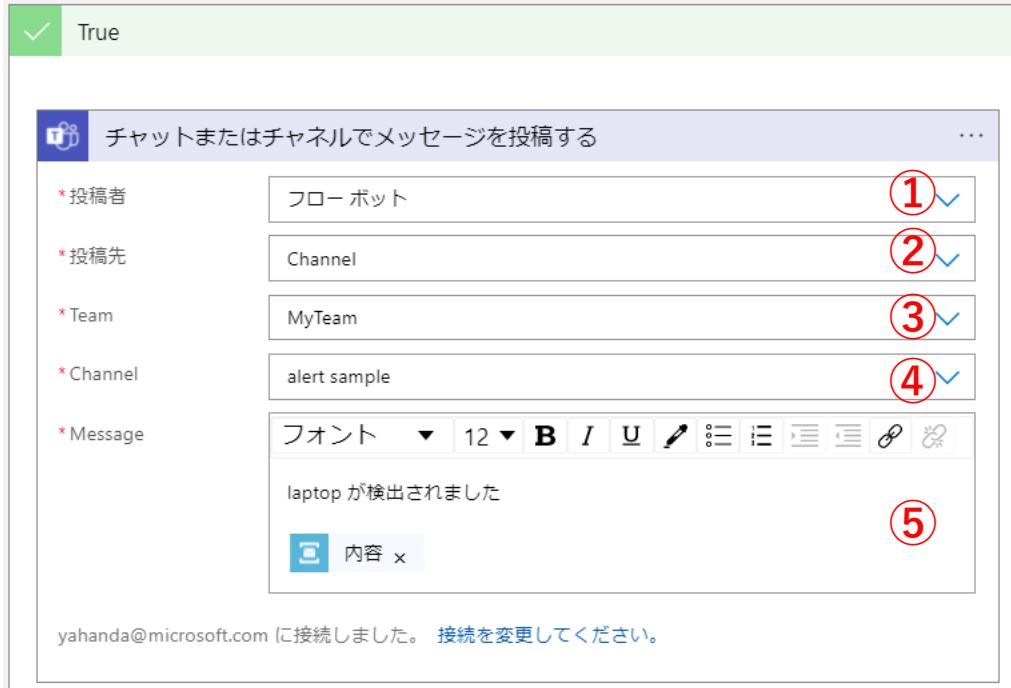
Step5: Teams アラートの実装 (Logic Apps)

- 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]を選択します。
- ② [アクション]で、[チャットまたはチャネルでメッセージを投稿する]を選択します。

※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



Step5: Teams アラートの実装 (Logic Apps)



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 :

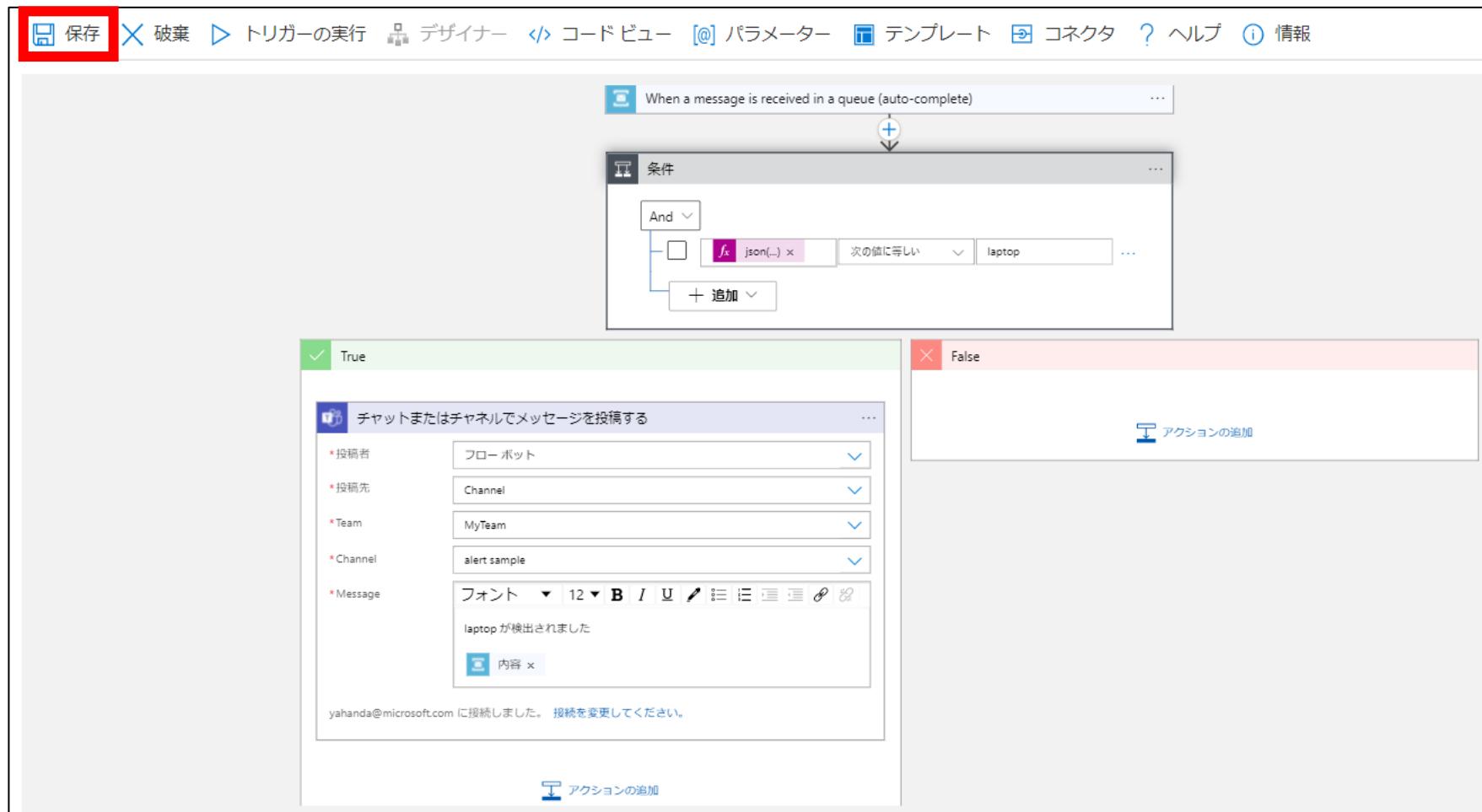
※[動的なコンテンツの追加] > [内容] より Stream Analytics から送信されたメッセージのコンテンツが追加できます

Step5: Teams アラートの実装 (Logic Apps)

[保存]を選択して、変更内容を保存します。

Teams アラートの実装の設定は以上になります。

カメラにアラート上げる対象の物体を映すとTEAMSに検出が投稿されます。



Appendix: 各種パラメータ設定

Azure Portal の IoT Hub の画面より、Percept DK 上で稼働するモジュールの各種パラメータ設定が可能です。
[IoT Edge] > 該当デバイスID > [モジュールを設定する] をクリックします。

The diagram illustrates the process of configuring a module on the Percept DK device via the Azure IoT Edge interface. It consists of two screenshots connected by a large grey arrow pointing from left to right.

Left Screenshot (Azure IoT Edge Device Settings):

- The title bar shows "iot-percept-demo | IoT Edge".
- The left sidebar menu has "IoT Edge" selected, highlighted with a red box.
- The main content area is titled "IoT Edge デバイス" and "IoT Edge デプロイ".
- Under "デバイス名", there is a "デバイス ID の入力" field and a "デバイスの検索" button.
- A table lists devices with columns "デバイス ID" and "ランタイムの応答". The row for "percept01" has its "デバイス ID" cell highlighted with a red box.
- At the bottom, there are buttons for "IoT Edge デバイスの追加", "最新の情報に更新", and "削除".

Right Screenshot (Module Configuration Page):

- The title bar shows "percept01" under "iot-percept-demo".
- The top navigation bar includes "モジュールを設定する" (highlighted with a red box), "子デバイスを管理する", "トラブルシューティング", and "デバイスツイン".
- The main content area displays device details:
 - デバイス ID: percept01
 - 主キー: (redacted)
 - セカンダリキー: (redacted)
 - プライマリ接続文字列: (redacted)
 - セカンダリ接続文字列: (redacted)
 - IoT Edge ランタイムの応答: 200 -- OK
 - IoT Hub への接続を有効にする: 有効化 無効化
 - 親デバイス: (redacted)

Appendix: 各種パラメータ設定

[azureeyemodule] > [モジュールツインの設定] より各種パラメータ設定が可能です。

設定後、[更新する] > [確認および作成] > [作成] にて設定情報が Percept DK デバイス側に反映されます。

デバイスのモジュールを設定してください: percept01

iot-percept-demo

モジュール ルート 確認および作成

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 コンテナーの作成オプション **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/tiny-yolo-v2.zip",  
4   "RawStream": true,  
5   "ResultStream": true,  
6   "Running": true,  
7   "TelemetryIntervalNeuralNetworkMs": 3000  
8 }
```

更新する キャンセル

Appendix: 各種パラメータ設定

主要なパラメータは以下の通りです。その他パラメータ詳細については、以下リンクを参照ください。

<https://github.com/microsoft/azure-percept-advanced-development/tree/main/azureeyemodule#module-twin-values>

モジュールツイン	データ型	説明	設定例
Logging	Boolean	デバッグルогの有効化/無効化	"Logging": true
ModelZipUrl	String	モデルのURL ※既成のAIモデルとして利用可能なモデルURLは以下を参照 https://github.com/microsoft/azure-percept-advanced-development#model-urls	"ModelZipUrl": "https://aedsamples.blob.core.windows.net/vision/aeddevkitnew/openpose.zip"
RawStream	Boolean	Rawカメラフィードのストリーム有効化/無効化	"RawStream": true
ResultStream	Boolean	推論結果（Rawカメラフィード + 検出結果のマーカアップ）のストリーム有効化/無効化	"ResultStream": true
TelemetryInterval NeuralNetworkMs	Integer	推論結果の送信頻度（ミリ秒単位で指定）	"TelemetryIntervalNeuralNetworkMs": 3000