**1)Export data from Azure IoT Central and visualize insights in Power BI**

-Configure an IoT Central application to export telemetry to an event hub.

-Use Logic Apps to send data from an event hub to a Power BI streaming dataset.

-Create a Power BI dashboard to visualize data in the streaming dataset.

**1.1)Create an event hub**

-In the Azure portal,select the Event Hubs page, select Create.

-On the Create Namespace page, take the following steps:

-Enter a unique name for the namespace such as yourname-retail-store-analysis. The system checks to see if this name is available.

-Choose the Basic pricing tier.

-Select the same Subscription you used to create your IoT Central application.

-Select the retail-store-analysis resource group.

-Select the same location you used for your IoT Central application.

-Select Create. You may have to wait a few minutes for the system to provision the resources.

Now you have an Event Hubs Namespace, you can create an Event Hub to use with your IoT Central application:

On the home page for your Event Hubs Namespace in the portal, select + Event Hub.

On the Create Event Hub page, enter store-telemetry as the name, and then select Create.

You now have an event hub you can use when you configure data export from your IoT Central application.

**1.2)Configure data export**

Now you have an event hub, you can configure your In-store analytics - checkout application to export telemetry from the connected devices. The following steps show you how to configure the export:

Sign in to your In-store analytics - checkout IoT Central application.

Select Data export in the left pane.

Select New > Azure Event Hubs.

Enter Telemetry export as the Display Name.

Select your Event Hubs namespace.

Select the store-telemetry event hub.

Switch off Devices and Device Templates in the Data to export section.

Select Save.

The data export may take a few minutes to start sending telemetry to your event hub. You can see the status of the export on the Data exports page.

**1.3)Create the Power BI datasets**

Your Power BI dashboard will display data from your retail monitoring application. In this solution, you use Power BI streaming datasets as the data source for the Power BI dashboard. In this section, you define the schema of the streaming datasets so that the logic app can forward data from the event hub. The following steps show you how to create two streaming datasets for the environmental sensors and one streaming dataset for the occupancy sensor:

-Sign in to your Power BI account.

-Select Workspaces, and then select Create a workspace.

-On the Create a workspace page, enter In-store analytics - checkout as the Workspace name.

-Scroll to the bottom of the Welcome to the In-store analytics - checkout workspace page, and select Skip.

-On the workspace page, select Create > Streaming dataset.

-On the New streaming dataset page, choose API, and then select Next.

-Enter Zone 1 sensor as the Dataset name.

-Enter the three Values from stream in following table:

TABLE 1

Value name Value type

Timestamp DateTime

Humidity Number

Temperature Number

-Switch Historic data analysis on.

-Select Create and then Done.

-Create another streaming dataset called Zone 2 sensor with the same schema and settings as the Zone 1 sensor streaming dataset.

You now have two streaming datasets. The logic app will route telemetry from the two environmental sensors connected to your In-store analytics - checkout application to these two datasets:

This solution uses one streaming dataset for each sensor because it's not possible to apply filters to streaming data in Power BI.

You also need a streaming dataset for the occupancy telemetry:

-On the workspace page, select Create > Streaming dataset.

-On the New streaming dataset page, choose API, and then select Next.

-Enter Occupancy sensor as the Dataset name.

-Enter the five Values from stream in following table:

TABLE 2

Value name Value type

Timestamp DateTime

Queue Length 1 Number

Queue Length 2 Number

Dwell Time 1 Number

Dwell Time 2 Number

-Switch Historic data analysis on.

-Select Create and then Done.

You now have a third streaming dataset that stores values from the simulated occupancy sensor. This sensor reports the queue length at the two checkouts in the store, and how long customers are waiting in these queues:

**1.4)Create a logic app**

In this solution, the logic app reads telemetry from the event hub, parses the data, and then sends it to the Power BI streaming datasets you created.

Before you create the logic app, you need the device IDs of the two RuuviTag sensors you connected to your IoT Central application.

-Sign in to your In-store analytics - checkout IoT Central application.

-Select Devices in the left pane. Then select RuuviTag.

-Make a note of the Device IDs.

The following steps show you how to create the logic app in the Azure portal:

-Sign in to the Azure portal and select Create a resource at the top left of the screen.

-In Search the Marketplace, enter Logic App, and then press Enter.

-On the Logic App page, select Create.

-On the Logic App create page:

-Enter a unique name for your logic app such as yourname-retail-store-analysis.

-Select the same Subscription you used to create your IoT Central application.

-Select the retail-store-analysis resource group.

-Select the same location you used for your IoT Central application.

-Select Create. You may have to wait a few minutes for the system to provision the resources.

-In the Azure portal, navigate to your new logic app.

-On the Logic Apps Designer page, scroll down and select Blank Logic App.

-In Search connectors and triggers, enter Event Hubs.

-In Triggers, select When events are available in Event Hub.

-Enter Store telemetry as the Connection name, and select your Event Hubs Namespace.

-Select the RootManageSharedAccess policy, and select Create.

-In the When events are available in Event Hub action:

-In Event Hub name, select store-telemetry.

-In Content type, select application/json.

-Set the Interval to three and the Frequency to seconds

-Select Save to save your logic app.

To add the logic to your logic app design, select Code view:

Replace "actions": {}, with the following JSON. Replace the two placeholders [YOUR RUUVITAG DEVICE ID 1] and [YOUR RUUVITAG DEVICE ID 2] with the IDs you noted of your two RuuviTag devices:

"actions": {

"Initialize\_Device\_ID\_variable": {

"inputs": {

"variables": [

{

"name": "DeviceID",

"type": "String"

}

]

},

"runAfter": {},

"type": "InitializeVariable"

},

"Initialize\_Interface\_ID\_variable": {

"inputs": {

"variables": [

{

"name": "InterfaceID",

"type": "String",

"value": "Other"

}

]

},

"runAfter": {

"Initialize\_Device\_ID\_variable": [

"Succeeded"

]

},

"type": "InitializeVariable"

},

"Parse\_Properties": {

"inputs": {

"content": "@triggerBody()?['Properties']",

"schema": {

"properties": {

"iothub-connection-auth-generation-id": {

"type": "string"

},

"iothub-connection-auth-method": {

"type": "string"

},

"iothub-connection-device-id": {

"type": "string"

},

"iothub-enqueuedtime": {

"type": "string"

},

"iothub-interface-name": {

"type": "string"

},

"iothub-message-source": {

"type": "string"

},

"x-opt-enqueued-time": {

"type": "string"

},

"x-opt-offset": {

"type": "string"

},

"x-opt-sequence-number": {

"type": "integer"

}

},

"type": "object"

}

},

"runAfter": {

"Initialize\_Interface\_ID\_variable": [

"Succeeded"

]

},

"type": "ParseJson"

},

"Parse\_Telemetry": {

"inputs": {

"content": "@triggerBody()?['ContentData']",

"schema": {

"properties": {

"DwellTime1": {

"type": "number"

},

"DwellTime2": {

"type": "number"

},

"count1": {

"type": "number"

},

"count2": {

"type": "number"

},

"humidity": {

"type": "number"

},

"temperature": {

"type": "number"

}

},

"type": "object"

}

},

"runAfter": {

"Initialize\_Interface\_ID\_variable": [

"Succeeded"

]

},

"type": "ParseJson"

},

"Set\_Device\_ID\_variable": {

"inputs": {

"name": "DeviceID",

"value": "@body('Parse\_Properties')?['iothub-connection-device-id']"

},

"runAfter": {

"Parse\_Properties": [

"Succeeded"

]

},

"type": "SetVariable"

},

"Set\_Interface\_ID\_variable": {

"inputs": {

"name": "InterfaceID",

"value": "@body('Parse\_Properties')?['iothub-interface-name']"

},

"runAfter": {

"Set\_Device\_ID\_variable": [

"Succeeded"

]

},

"type": "SetVariable"

},

"Switch\_by\_DeviceID": {

"cases": {

"Occupancy": {

"actions": {

"Switch\_by\_InterfaceID": {

"cases": {

"Dwell\_Time\_interface": {

"actions": {},

"case": "RS40\_Occupancy\_Sensor\_v2\_1l0"

},

"People\_Count\_interface": {

"actions": {},

"case": "RS40\_Occupancy\_Sensor\_iv"

}

},

"default": {

"actions": {}

},

"expression": "@variables('InterfaceID')",

"runAfter": {},

"type": "Switch"

}

},

"case": "Occupancy"

},

"Zone 2 environment": {

"actions": {},

"case": "[YOUR RUUVITAG DEVICE ID 2]"

},

"Zone\_1\_environment": {

"actions": {},

"case": "[YOUR RUUVITAG DEVICE ID 1]"

}

},

"default": {

"actions": {}

},

"expression": "@variables('DeviceID')",

"runAfter": {

"Parse\_Telemetry": [

"Succeeded"

],

"Set\_Interface\_ID\_variable": [

"Succeeded"

]

},

"type": "Switch"

}

},

-Select Save and then select Designer to see the visual version of the logic you added:

-Select Switch by DeviceID to expand the action. Then select Zone 1 environment, and select Add an action.

-In Search connectors and actions, enter Power BI, and then press Enter.

-Select the Add rows to a dataset (preview) action.

-Select Sign in and follow the prompts to sign in to your Power BI account.

-After the sign-in process is complete, in the Add rows to a dataset action:

--Select In-store analytics - checkout as the workspace.

--Select Zone 1 sensor as the dataset.

--Select RealTimeData as the table.

--Select Add new parameter and then select the Timestamp, Humidity, and Temperature fields.

--Select the Timestamp field, and then select x-opt-enqueuedtime from the Dynamic content list.

--Select the Humidity field, and then select See more next to Parse Telemetry. Then select humidity.

--Select the Temperature field, and then select See more next to Parse Telemetry. Then select temperature.

--Select Save to save your changes.

-Select the Zone 2 environment action, and select Add an action.

-In Search connectors and actions, enter Power BI, and then press Enter.

-Select the Add rows to a dataset (preview) action.

-In the Add rows to a dataset 2 action:

--Select In-store analytics - checkout as the workspace.

--Select Zone 2 sensor as the dataset.

--Select RealTimeData as the table.

--Select Add new parameter and then select the Timestamp, Humidity, and Temperature fields.

--Select the Timestamp field, and then select x-opt-enqueuedtime from the Dynamic content list.

--Select the Humidity field, and then select See more next to Parse Telemetry. Then select humidity.

--Select the Temperature field, and then select See more next to Parse Telemetry. Then select temperature. Select Save to save your changes.

-Select the Occupancy action, and then select the Switch by Interface ID action.

-Select the Dwell Time interface action, and select Add an action.

-In Search connectors and actions, enter Power BI, and then press Enter.

-Select the Add rows to a dataset (preview) action.

-In the Add rows to a dataset action:

--Select In-store analytics - checkout as the workspace.

--Select Occupancy Sensor as the dataset.

--Select RealTimeData as the table.

--Select Add new parameter and then select the Timestamp, Dwell Time 1, and Dwell Time 2 fields.

--Select the Timestamp field, and then select x-opt-enqueuedtime from the Dynamic content list.

--Select the Dwell Time 1 field, and then select See more next to Parse Telemetry. Then select DwellTime1.

--Select the Dwell Time 2 field, and then select See more next to Parse Telemetry. Then select DwellTime2.

--Select Save to save your changes.

-Select the People Count interface action, and select Add an action.

-In Search connectors and actions, enter Power BI, and then press Enter.

-Select the Add rows to a dataset (preview) action.

-In the Add rows to a dataset action:

--Select In-store analytics - checkout as the workspace.

--Select Occupancy Sensor as the dataset.

--Select RealTimeData as the table.

--Select Add new parameter and then select the Timestamp, Queue Length 1, and Queue Length 2 fields.

--Select the Timestamp field, and then select x-opt-enqueuedtime from the Dynamic content list.

--Select the Queue Length 1 field, and then select See more next to Parse Telemetry. Then select count1.

--Select the Queue Length 2 field, and then select See more next to Parse Telemetry. Then select count2.

--Select Save to save your changes.

The logic app runs automatically. To see the status of each run, navigate to the Overview page for the logic app in the Azure portal:

**1.5)Create a Power BI dashboard**

Now you have telemetry flowing from your IoT Central application through your event hub. Then your logic app parses the event hub messages and adds them to a Power BI streaming dataset. Now, you can create a Power BI dashboard to visualize the telemetry:

Sign in to your Power BI account.

Select Workspaces > In-store analytics - checkout.

Select Create > Dashboard.

Enter Store analytics as the dashboard name, and select Create.

-Add line charts

Add four line chart tiles to show the temperature and humidity from the two environmental sensors. Use the information in the following table to create the tiles. To add each tile, start by selecting ...(More options) > Add Tile. Select Custom Streaming Data, and then select Next:

ADD LINE CHARTS

Setting Chart #1 Chart #2 Chart #3 Chart #4

Dataset Zone 1 sensor Zone 1 sensor Zone 2 sensor Zone 2 sensor

Visualization type Line chart Line chart Line chart Line chart

Axis Timestamp Timestamp Timestamp Timestamp

Values Temperature Humidity Temperature Humidity

Time window 60 minutes 60 minutes 60 minutes 60 minutes

Title Temperature (1 hour) Humidity (1 hour) Temperature (1 hour) Humidity (1 hour)

Subtitle Zone 1 Zone 1 Zone 2 Zone 2

-Add cards to show environmental data

Add four card tiles to show the most recent temperature and humidity values from the two environmental sensors. Use the information in the following table to create the tiles. To add each tile, start by selecting ...(More options) > Add Tile. Select Custom Streaming Data, and then select Next:

ADD CARDS TO SHOW ENVIRONMENTAL DATA

Setting Card #1 Card #2 Card #3 Card #4

Dataset Zone 1 sensor Zone 1 sensor Zone 2 sensor Zone 2 sensor

Visualization type Card Card Card Card

Fields Temperature Humidity Temperature Humidity

Title Temperature (F) Humidity (%) Temperature (F) Humidity (%)

Subtitle Zone 1 Zone 1 Zone 2 Zone 2

-Add tiles to show checkout occupancy data

Add four card tiles to show the queue length and dwell time for the two checkouts in the store. Use the information in the following table to create the tiles. To add each tile, start by selecting ...(More options) > Add Tile. Select Custom Streaming Data, and then select Next:

ADD TILES TO SHOW CHECKOUT OCCUPANCY DATA

Setting Card #1 Card #2 Card #3 Card #4

Dataset Occupancy sensor Occupancy sensor Occupancy sensor Occupancy sensor

Visualization type Clustered column chart Clustered column chart Gauge Gauge

Axis Timestamp Timestamp N/A N/A

Value Dwell Time 1 Dwell Time 2 Queue Length 1 Queue Length 2

Time window 60 minutes 60 minutes N/A N/A

Title Dwell Time Dwell Time Queue Length Queue Length

Subtitle Checkout 1 Checkout 2 Checkout 1 Checkout 2