

Capture alerts in Dynamics

Exercise 4



Disconnected Simulator

The Simulator is currently not connected to any IoT Device or IoT Hub. Notice the "Connection Status: Disconnected" in the top right.

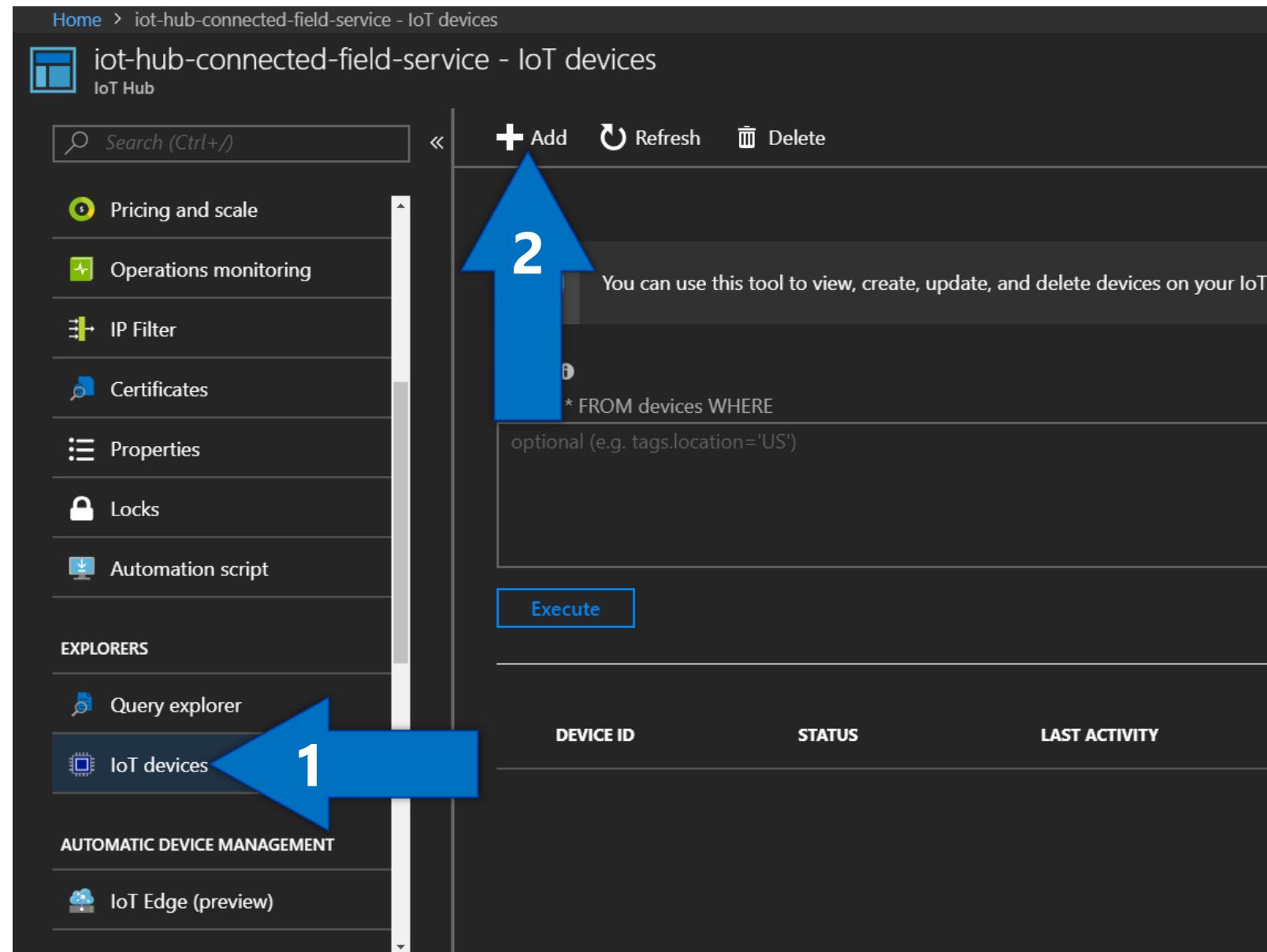
Without this connection no temperature or humidity data from this Simulator will be sent to Dynamics.

The screenshot shows a web-based interface for a Field Service IoT application. At the top, there's a navigation bar with tabs: 'Field Service IoT' (selected), 'Thermometer', 'Connection', and 'Reboot'. To the right of the tabs, the text 'Connection Status: Disconnected' is displayed in red. Below the navigation bar is a photograph of a digital thermometer device mounted on a wall. A finger is pointing at the device's screen, which displays the number '25.3' and the text '18.0 °C'. To the right of the photograph, there are two circular gauge charts. The first chart, labeled 'Humidity', shows a value of '40%' with a blue arc. The second chart, labeled 'Temperature', shows a value of '65°F' with a green arc. Above the charts are dropdown menus for 'Device ID' and a 'Refresh' button. At the bottom of the interface, there are two large green rectangular boxes labeled 'Messages Received:' and 'Messages Sent:'.

Add Simulator Device to IoT Hub

Head back to your Azure portal and open your IoT Hub.

- `1` Under the EXPLORERS section click IoT devices
- `2` Click Add

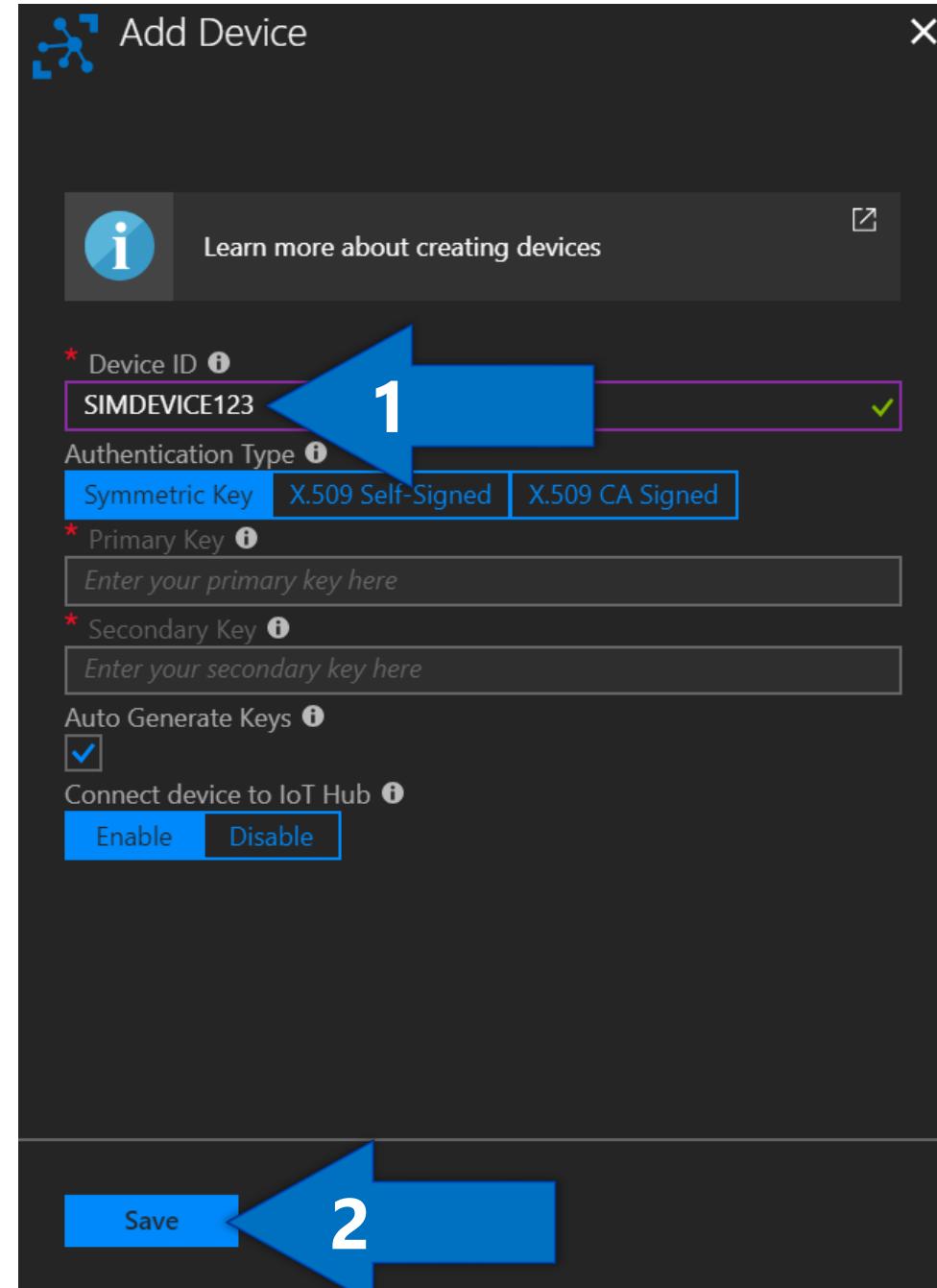


Simulator Device Information

The device we are adding to the IoT Hub is going to represent the device we use to communicate with the Thermostat Simulator.

In the Add Device form:

- `1` Give a unique Device ID, in my case SIMDEVICE123
- `2` Click Save



Azure connection information

In the last exercise we captured this information. Let's ensure we have captured it correctly.

- `1` We need the Hostname for our IoT Hub from the Overview section, Click to copy
- `2` Also go into Shared access policies and click the iothubowner Policy. Click to copy the Primary key

The screenshot shows two main sections of the Azure IoT Hub interface:

Top Section (Overview):

- Hostname:** iot-hub-connected-field-service.azure-devices.net
- Pricing and scale tier:** S1 - Standard
- Number of IoT Hub units:** 1

A blue arrow points from the "Hostname" text to a "Click to copy" button.

Bottom Section (Shared access policies):

The "iot-hubowner" policy is selected. The "Permissions" section includes checked boxes for Registry read, Registry write, Service connect, and Device connect.

The "Shared access keys" section displays the Primary key and Secondary key, each with a "Copy" icon. A large blue arrow labeled "2" points to the "Primary key" field.

The left sidebar shows other tabs like Overview, Activity log, and Access control (IAM).



Disconnected Simulator

Head back to the tab your simulator is open in and click Connection

The screenshot shows a web-based interface for a Field Service IoT application. At the top, there is a navigation bar with tabs: 'Field Service IoT' (selected), 'Thermometer', 'Connection', and 'Reboot'. To the right of the tabs, it displays 'Connection Status: Disconnected'. Below the navigation bar is a photograph of a digital thermometer device mounted on a wall. A blue arrow points upwards from the text instructions to the 'Connection' tab in the navigation bar. On the right side of the interface, there is a 'Device ID' dropdown menu and a 'Refresh' button with a red dot indicating an update. Below these are two circular gauge charts: one for 'Humidity' showing 40% and another for 'Temperature' showing 65°F. At the bottom, there are two large green rectangular boxes labeled 'Messages Received:' and 'Messages Sent:'.

Connection Status: Disconnected

Device ID •

Humidity 40%

Temperature 65°F

Messages Received:

Messages Sent:

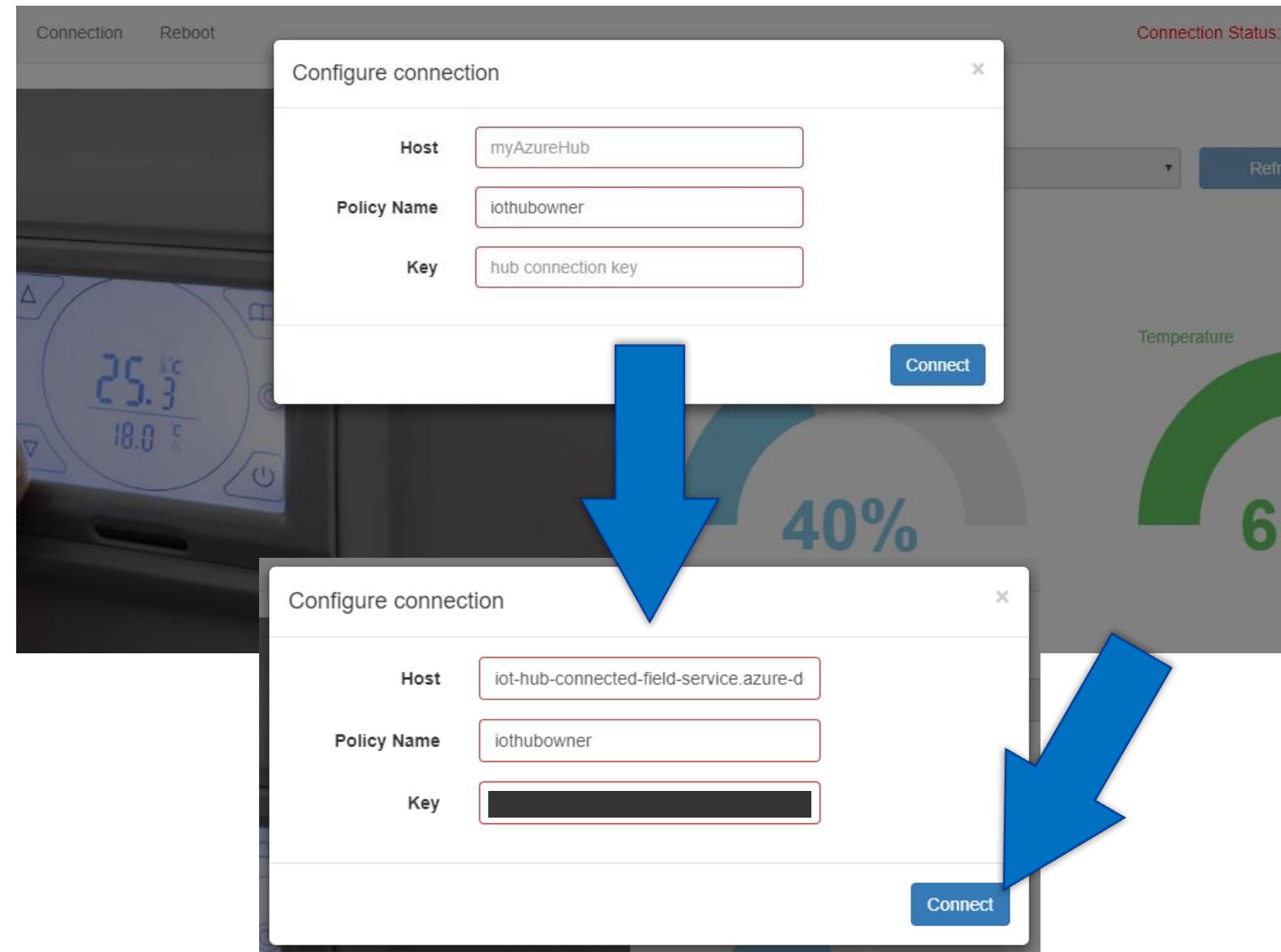
Configure connection

Paste the Hostname you copied to the Host field.

Paste iothubowner to the Policy name field.

Paste the Primary key for the iothubowner shared access policy in the Key field.

Click connect when complete.



Link simulator to device

The Simulator is now connected to your IoT Hub. Notice the "Connection Status: connected" in the top right.

Click the Device ID drop down and the select the simulator device we created earlier. In my case it is SIMDEVICE123.

The screenshot shows the Field Service IoT interface. At the top, there are tabs for "Field Service IoT", "Thermometer", "Connection", and "Reboot". The "Connection" tab is active. To the right, the "Connection Status" is shown as "Connected". Below the tabs, there is a photograph of a digital thermometer device mounted on a wall. A hand is pointing at the device's screen, which displays the number "25.3". A large blue arrow points from the device towards a dropdown menu labeled "Device ID". The dropdown menu has three options: "Select a device", "Select a device", and "SIMDEVICE123", with "SIMDEVICE123" being the selected option. To the right of the dropdown is a "Refresh" button with a red dot indicating updates. Below the device image, there are two circular gauge-like indicators. The left one is labeled "Humidity" and shows "40%". The right one is labeled "Temperature" and shows "65°F". At the bottom, there are two sections: "Messages Received:" and "Messages Sent:", each with a long green progress bar.

Field Service IoT Thermometer Connection Reboot

Connection Status: Connected

Device ID

Select a device Select a device SIMDEVICE123 Refresh

Humidity 40%

Temperature 65°F

Messages Received:

Messages Sent:

Automatically messages are sent

Now Humidity and Temperature messages will be sent from this Thermostat simulator directly to our simulator device in our IoT Hub.

If you ever want to stop sending messages click the Refresh button.

The screenshot shows the Field Service IoT interface. At the top, there are tabs for "Field Service IoT", "Thermometer", "Connection", and "Reboot". The "Connection" tab is selected, indicated by a grey background. To the right, it says "Connection Status: Connected". Below the tabs, there's a "Device ID" dropdown set to "SIMDEVICE123" with a refresh button next to it. A green dot indicates the connection is active. On the left, a photograph shows a hand interacting with a digital thermostat device. The device's screen displays "25.3 °C" and "18.0 °C". To the right, two circular gauges show "Humidity" at 40% (blue) and "Temperature" at 65°F (green). Below these are sections for "Messages Received:" and "Messages Sent:". The "Messages Received:" section is empty. The "Messages Sent:" section lists five entries: "CURRENT STATUS -> Te", "CURRENT STATUS -> Temperature: 65 Humidity: 40", "CURRENT STATUS -> Temperature: 65 Humidity: 40", "CURRENT STATUS -> Temperature: 65 Humidity: 40", and "CURRENT STATUS -> Temperature: 65 Humidity: 40".

Field Service IoT Thermometer Connection Reboot

Connection Status: Connected

Device ID: SIMDEVICE123 Refresh

Humidity: 40%

Temperature: 65°F

Messages Received:

Messages Sent:

- CURRENT STATUS -> Te
- CURRENT STATUS -> Temperature: 65 Humidity: 40

Data feed into IoT Hub

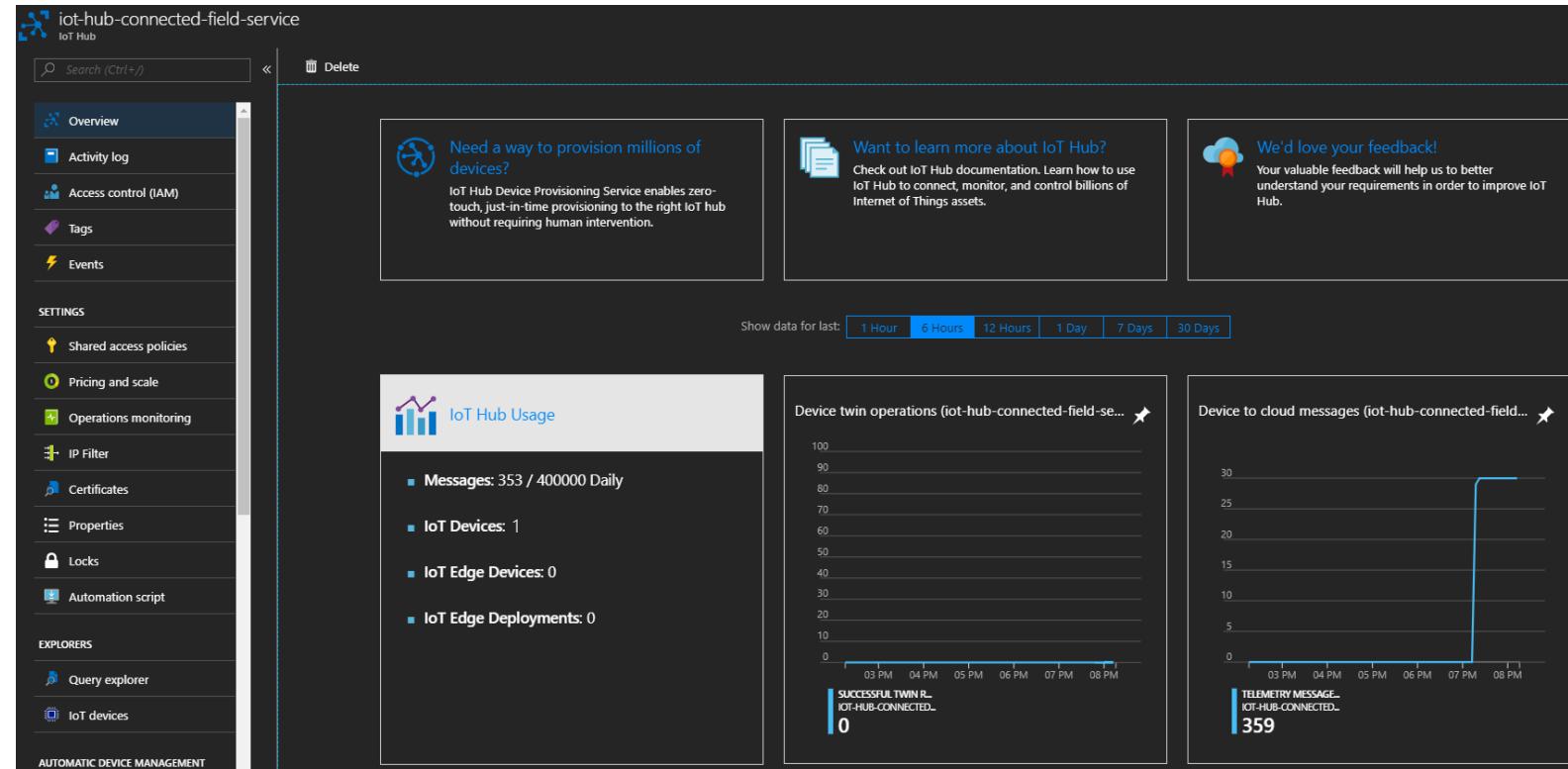
Where are our messages going?

How are these messages sent to Dynamics, they must start in Azure somewhere?

First visit your IoT Hub.

Our IoT Hub serves as the bi directional messaging infrastructure for our simulator. Right now it's doing the work of ingesting the Humidity and Temperature messages our device is sending.

I left my Simulator running for one hour and the resulting messaging data is seen here.



IoT data using Stream Analytics

Navigate to your Stream Analytics Job with the Alerts Queue Output. As an input. In my case it is rgconnectedfieldservice96e5071ebdd64bf88.

Our job takes our IoT Stream (Temperature and Humidity data) as input and a DevicesRulesBlob (contains a JSON payload defining our temperature threshold for when to send Alerts)

Our job sends a message to the AlertsQueue based on our Query and the threshold from our DevicesRulesBlob.

The screenshot shows the Azure Stream Analytics job overview page for the job `rgconnectedfieldservice96e5071ebdd64bf88`. The job status is **Stopped**. Key details include:

- Resource group:** rg-connected-field-service
- Status:** Stopped
- Location:** West US
- Subscription:** Super Saiyan Goku
- Subscription ID:** 6e1349e1-005a-44ba-8190-1013f89aab24
- Created:** Wednesday, February 28, 2018, 11:57:36 AM
- Started:** Thursday, April 26, 2018, 12:08:13 PM
- Last output:** -
- Hosting environment:** Cloud

Inputs:

- 2 DeviceRulesBlob
- 1 IoTStream

Outputs:

- 1 AlertsQueue

Query:

```

1 WITH AlertData AS
2 (
3     SELECT
4         Stream.DeviceID,
5         'Temperature' AS ReadingType,
6         Stream.Temperature AS Reading,
7         Stream.EventToken AS EventToken,
8         Ref.Temperature AS Threshold,
9         Ref.TemperatureRuleOutput AS RuleOutput,
10        Stream.EventEnqueuedUtcTime AS [Time]
11    FROM IoTStream Stream
12    JOIN DeviceRulesBlob Ref ON Ref.DeviceType = 'Thermostat'
13    WHERE
14        Ref.Temperature IS NOT null AND Stream.Temperature > Ref.

```

Monitoring:

Blob Storage

We've visited our IoT Hub and have a good picture of how this input functions.

Now visit your Storage account (in my cases called `storageconnfs`) to see where the `DevicesRulesBlob` and JSON lives for our Alerts threshold.

Click on the Blobs service and click into the `devicerules` container. You will need to click into the container folders

2016-05-30

> 22-40

Here you will see the `devicerules` JSON

The screenshot shows the Azure Storage account interface for the account `storageconnfs`. The left sidebar lists various services: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Storage Explorer (preview), SETTINGS, Access keys, and Configuration. The `Blobs` service is selected, indicated by a blue arrow. The main content area shows the `Blob service` for the `storageconnfs` account with a `Container` button, Refresh, and Delete options. The `devicerules` container is selected in the list, shown in the `Container` view with an `Overview` tab selected. The Storage Explorer (preview) view shows the contents of the `devicerules` container, including a file named `devicerules.json`.

Stream Analytics Alert Threshold

This is a deeper look at our stream analytics job query and the device rules JSON.

Notice from this we can tell our temperature threshold for this Connected Field Service Add-on solution is 70.0°F out of the box.

And this threshold is referenced in the JOIN to the DevcieRulesBlobm, allowing us to use the WHERE clause defining an alert as a temperature from our IoT Hub data stream greater than the threshold defined in the device rules JSON.

Stream.Temperature > Ref.Temperature

STREAM ANALYTICS QUERY

```
WITH AlertData AS
(
SELECT
    Stream.DeviceID,
    'Temperature' AS ReadingType,
    Stream.Temperature AS Reading,
    Stream.EventToken AS EventToken,
    Ref.Temperature AS Threshold,
    Ref.TemperatureRuleOutput AS RuleOutput,
    Stream.EventEnqueuedUtcTime AS [Time]
FROM IoTStream Stream
JOIN DeviceRulesBlob Ref ON Ref.DeviceType = 'Thermostat'
WHERE
    Ref.Temperature IS NOT null AND Stream.Temperature > Ref.Temperature
)

SELECT data.DeviceId,
    data.ReadingType,
    data.Reading,
    data.EventToken,
    data.Threshold,
    data.RuleOutput,
    data.Time
INTO AlertsQueue
FROM AlertData data
WHERE LAG(data.DeviceID) OVER (PARTITION BY data.DeviceId, data.Reading,
    data.ReadingType LIMIT DURATION(minute, 1)) IS NULL
```

DEVICE RULES JSON

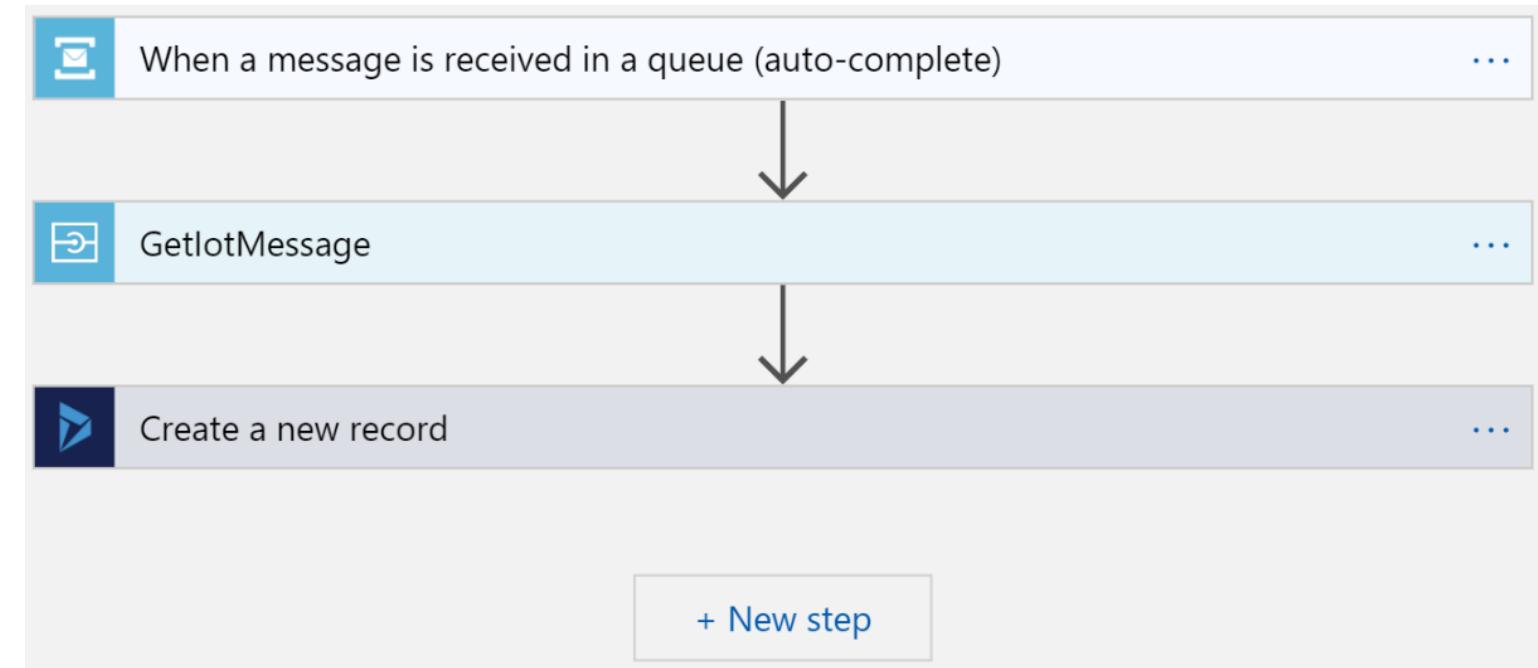
```
[
  {
    "DeviceType": "Thermostat",
    "Temperature": 70.0,
    "Humidity": null,
    "TemperatureRuleOutput": "AlarmTemp",
    "HumidityRuleOutput": null
  }
]
```



IoT to CRM Logic App

This Logic app is responsible for capturing the alert event from Stream Analytics and sending the necessary information to Dynamics 365.

This information is sent in the form of a new custom Dynamics Entity called IoT Alerts.



Start API App Queue Message Parser

The second step in our logic depends on one on the Queue Message Parser ur API Apps, so ensure it is running before continuing.

Go to the Queue Message Parser API App and click Start.

In my case my app is called QueueMessageParserrgconnectedfieldse rvicesc0e4e.

rg-connected-field-service

GetlotMessage

Subscription (change)
Super Saiyan Goku

Tags (change)
Click here to add tags

Filter by name... All types All locations

SETTINGS

- Quickstart
- Resource costs
- Deployments
- Policies
- Properties
- Locks

17 items Show hidden types

NAME ↑	TYPE ↑↓
iot-hub-connected-field-service	IoT Hub
IoTHubrgconnectedfieldservice6f633b0236b14f838	App Service
{IoT-To-CRM}	Logic app
QueueMessageParserrgconnectedfieldservicec0e4e	App Service
rgconnectedfieldservice5f94d97edd1540449	Stream Analytics job
rgconnectedfieldservice96e5071ebdd64bf88	Stream Analytics job
SBMessage	API Connection

QueueMessageParserrgconnectedfieldservicec0e4e

App Service

Browse **Start** **Swap**

Start Stream Analytics

Go back to your Stream Analytics Job with the AlertsQueue output.

Click the Start button in the Overview section and in the blade that appears click the Start button there too.

In my case the job is called rgconnectedfieldservice96e5071ebdd64bf88.

The screenshot shows the Azure Stream Analytics job overview page for the job 'rgconnectedfieldservice96e5071ebdd64bf88'. The left sidebar contains navigation links: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, SETTINGS (Locks), JOB TOPOLOGY (Inputs, Functions, Query, Outputs), CONFIGURE (Scale, Locale, Event ordering), Monitoring, and a large blue arrow pointing to the 'Start' button in the Monitoring blade.

The main content area displays the job details:

- rgconnectedfieldservice96e5071ebdd64bf88 Stream Analytics job**
- Start**, **Stop**, **Delete** buttons
- Inputs**: 2 (DeviceRulesBlob, IoTStream)
- Outputs**: 1 (AlertsQueue)
- Monitoring** blade: 'Start job' button, 'Job output start time' dropdown with 'Now' selected, and a large blue arrow pointing to the 'Start' button in this blade.
- Query** pane (partial view):


```

1 WITH AlertData AS
2 (
3     SELECT
4         Stream.DeviceID,
5             'Temperature' AS ReadingType,
6             Stream.Temperature AS Reading,
7             Stream.EventToken AS EventToken,
8             Ref.Temperature AS Threshold,
9             Ref.TemperatureRuleOutput AS RuleOutput,
10            Stream.EventEnqueuedUtcTime AS [Time]
11    FROM IoTStream Stream
12    JOIN DeviceRulesBlob Ref ON Ref.DeviceType = 'Thermostat'
13    WHERE
14        Ref.Temperature IS NOT null AND Stream.Temperature > Ref.
      
```
- Job Metrics** (partial view):

Send feedback
UserVoice
Created
Wednesday, February 28, 2018, 11:57:36 AM
Started
Thursday, April 26, 2018, 12:08:13 PM
Last output
-
Hosting environment
Cloud

Simulator temperature

Got back to the simulator web app and turn the temperature past the 70 degree threshold.

The screenshot shows a web-based interface for a Field Service IoT Thermometer. At the top, there's a navigation bar with tabs: 'Field Service IoT' (selected), 'Thermometer', 'Connection', and 'Reboot'. To the right of the tabs, it says 'Connection Status: Connected'. Below the navigation is a photograph of a digital wall-mounted thermometer displaying 25.3°C and 18.0°F. A hand is pointing at the screen. To the right of the device are two circular gauge charts: one for Humidity (40%) and one for Temperature (72°F). Below the device and charts are sections for 'Messages Received:' and 'Messages Sent:', each containing a single green message box. The 'Messages Sent:' box contains the text 'CURRENT STATUS -> Temperature: 72 Humidity: 40'.

Field Service IoT Thermometer Connection Reboot

Connection Status: Connected

Device ID: SIMDEVICE123 Refresh

Humidity: 40%

Temperature: 72°F

Messages Received:

Messages Sent:

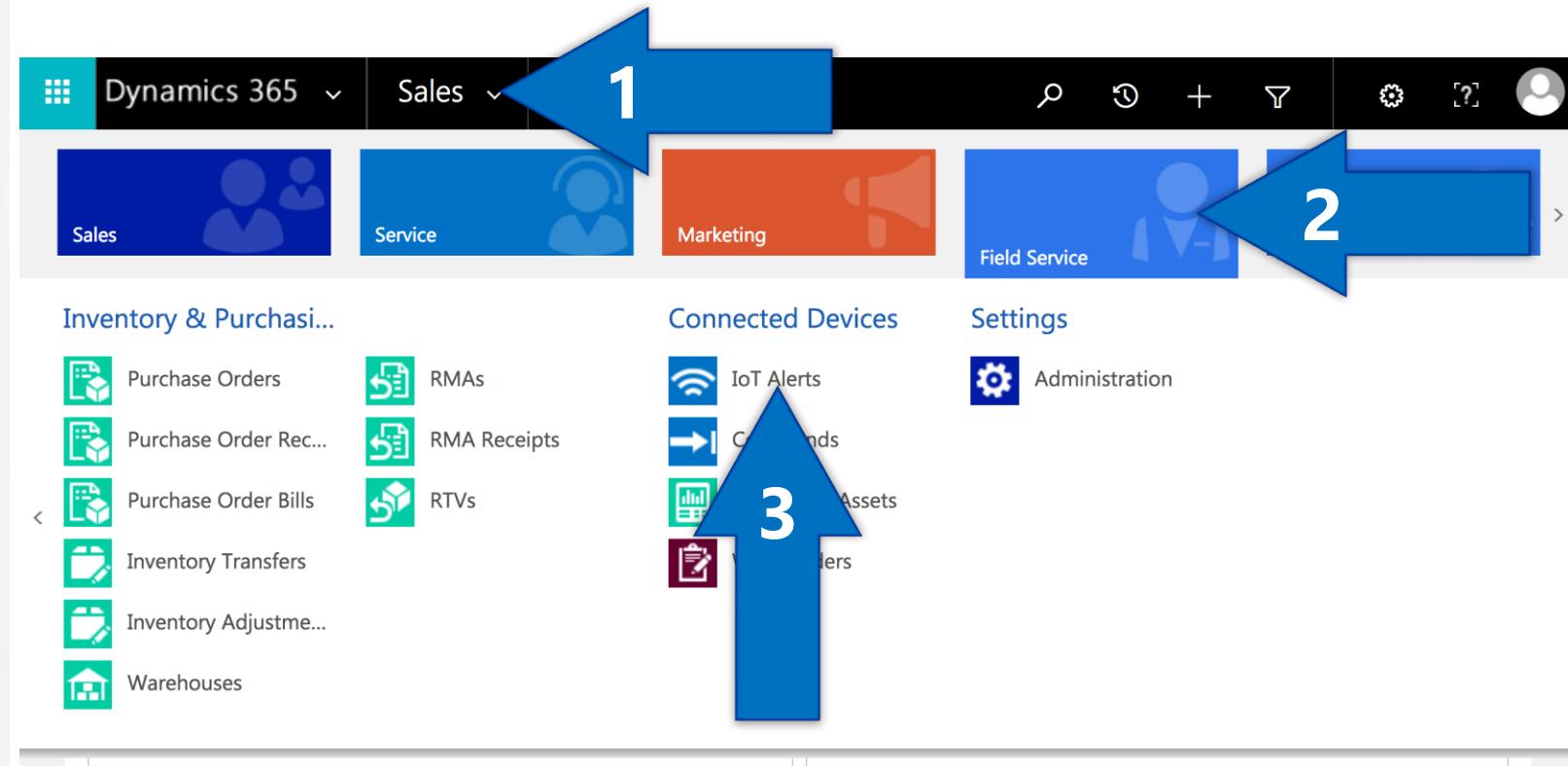
CURRENT STATUS -> Temperature: 72 Humidity: 40

Dynamics 365

In your browser go back to Dynamics 365 at your organization URL or just click on the tabs it's in if you haven't closed it.

The URL should look something like this
<https://{{YOUR ORGANIZATION NAME HERE}}.crm.dynamics.com/main.aspx>

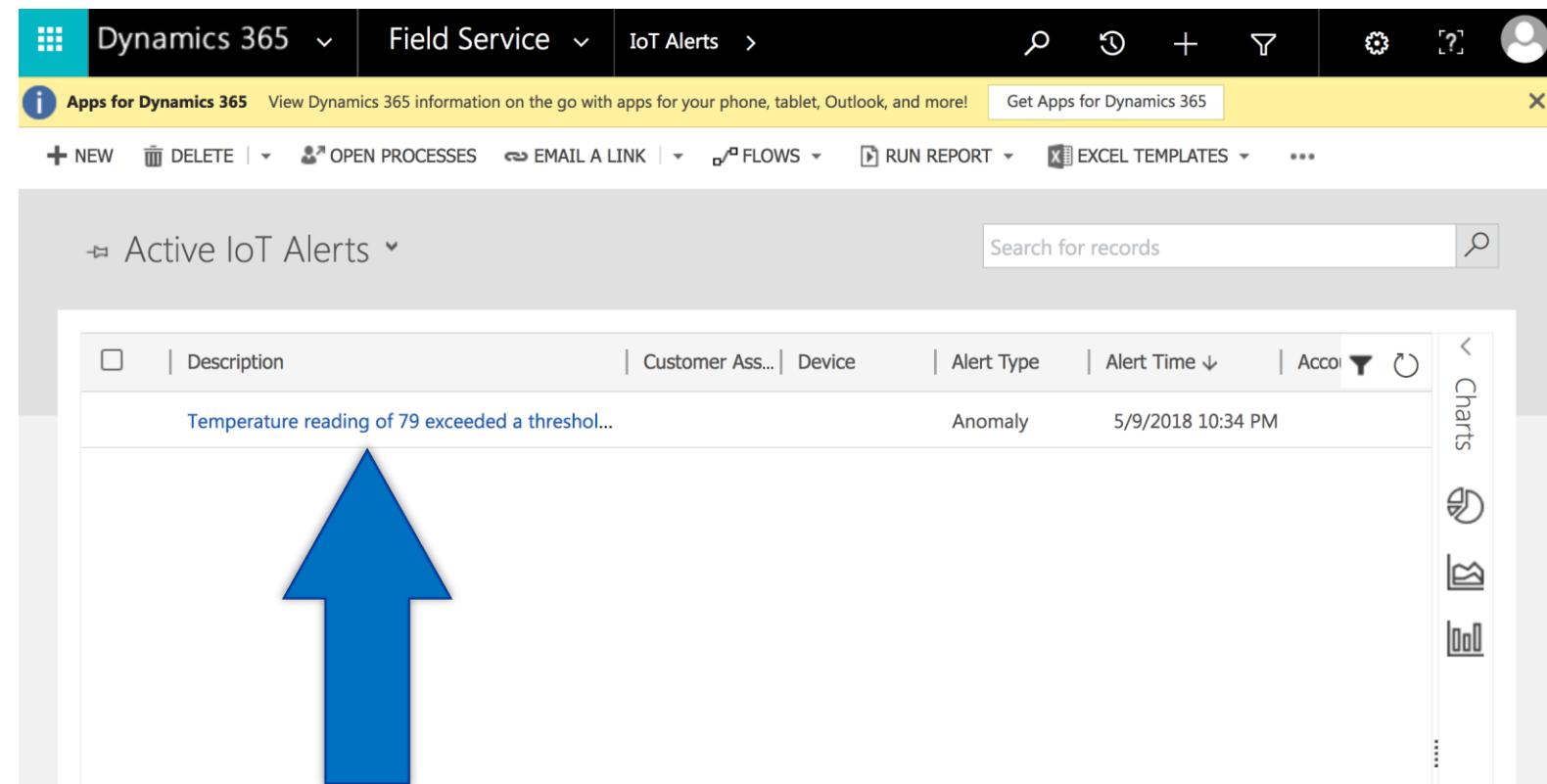
- `1` On the navigation bar click on the drop down next to Sales.
- `2` The Sitemap will appear, click Field Service.
- `3` Click on IoT Alerts.



Basic IoT Alert

Nice! We our first basic IoT Alert captured. Click into it.

If you visit the IoT-To-CRM logic you'll also notice a new succeeded Run. If you click it into you can see run details step by step in the logic app as well as the data piped through each step.



A screenshot of the Dynamics 365 Field Service - IoT Alerts page. The page title is "Active IoT Alerts". A single alert is listed in the grid:

Description	Customer Ass...	Device	Alert Type	Alert Time	Account
Temperature reading of 79 exceeded a threshold.			Anomaly	5/9/2018 10:34 PM	

A large blue arrow points upwards towards the alert description.

IoT Alert Information

We'll see the alert description for the reason this anomaly was detected, timestamp, and alert data in the form of JSON (it was thanks to https://en.wikipedia.org/wiki/Douglas_Crockford and many others for making this standard possible)

This new IoT Alert Entry still has the incomplete Customer Asset information. This is because the simulator device we registered in our IoT Hub was not registered as a connected device in Dynamics.

The Customer Asset is identified by Device ID, but there is none in the Dynamics system.

The screenshot shows the Dynamics 365 interface for managing IoT alerts. At the top, a navigation bar includes options like NEW, DEACTIVATE, DELETE, OPEN PROCESSES, CREATE COMMAND, PROCESS, ADD TO QUEUE, QUEUE ITEM DETAILS, ASSIGN, and more. The current item is 'IOT ALERT : INFORMATION'.

A message at the top states: "Temperature reading of 79 exceeded a threshold of 70". To the right, it shows the 'Owner' as 'Greg Degruy'.

The main area displays a process flow titled 'CFS - IoT Alert Process Flow (Active for 9 minutes)'. The steps shown are: Created (Active for 9 minutes) → Create Case → Create Work Order → Schedule Work Order → Close Work Order. The 'Create Case' step is currently active.

Below the process flow, there's a 'General' section with the following data:

- Description:** Temperature reading of 79 exceeded a threshold of 70
- Alert Type:** Anomaly
- Alert Token:** 82a40d4f-1c9e-4159-83b7-b9be078b0ad7
- Alert Time:** 5/9/2018 10:34 PM
- Alert URL:** (link)
- Alert Data:** A JSON object containing device information, reading type, temperature, event token, rule output, and time.

To the right of the general section are three panels: ASSISTANT, NOTES, and Connected Device Readings. The ASSISTANT panel contains a note input field with the placeholder 'Enter a note'. The NOTES panel displays a message: 'There aren't any notes to show. To go back to the alert entry, click here.' A large blue arrow points downwards from the NOTES panel towards the Customer Asset section.

The bottom section is titled 'CUSTOMER ASSET' and lists the following fields:

- Customer Asset:** (empty)
- Device:** (empty)
- Device ID:** SIMDEVICE123

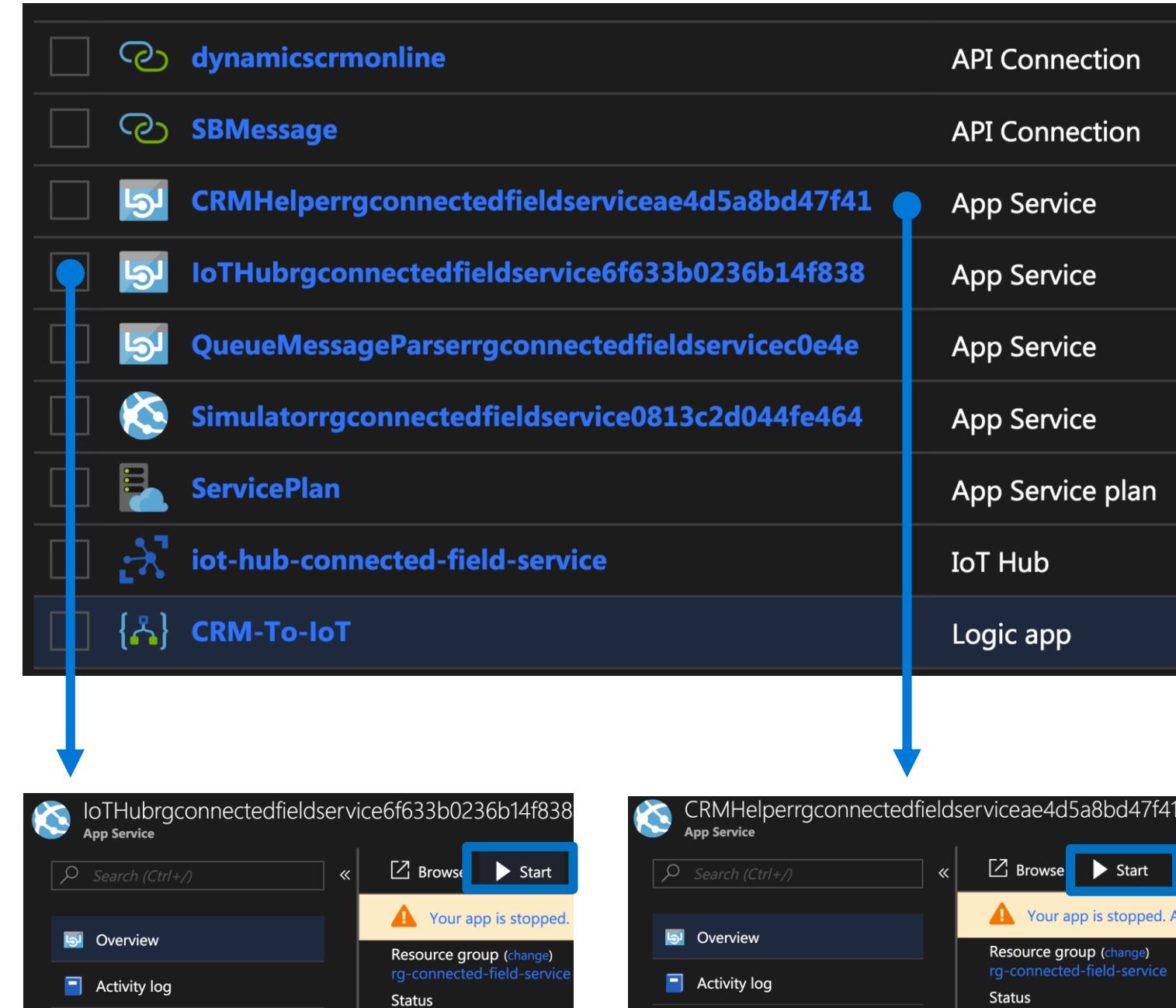
Start API Apps

Go back to your Azure portal and view your connected field service resource group. Our second logic app CRM-To-IoT is responsible for any Customer Asset registration in Dynamics.

This Logic App Depends on the CRMHelper and IoTHub Api apps, so we'll need to ensure they're running first.

Go to each API app and click start.

In my case I needed to start CRMHelperrgconnectedfieldserviceae4d5a8bd47f41 and IoTHubrgconnectedfieldservice6f633b0236b14f838 respectively.

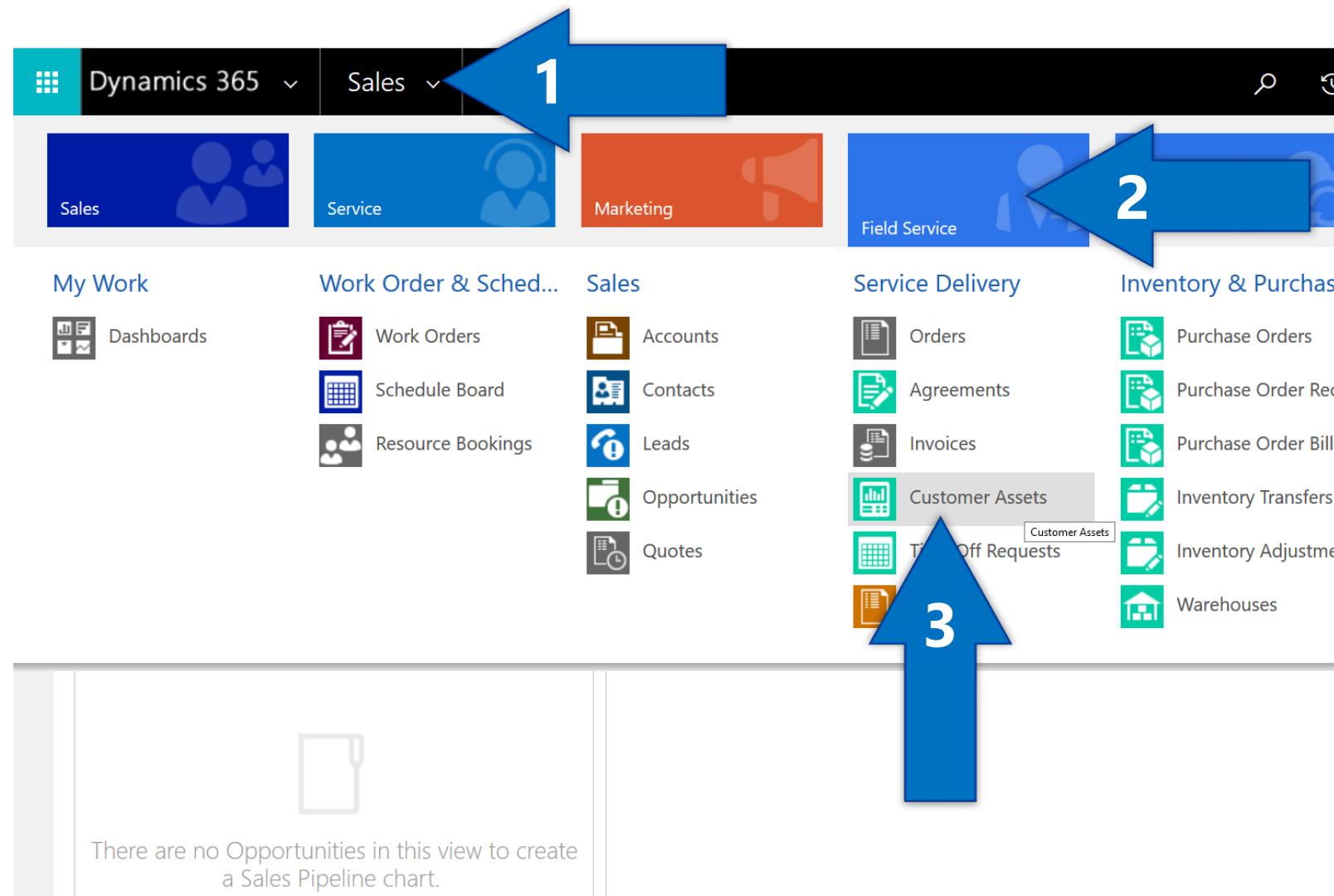


Add Dynamics Asset

In your browser go back to Dynamics 365 at your organization URL or just click on the tabs it's in if you haven't closed it.

The URL should look something like this
<https://{{YOUR ORGANIZATION NAME HERE}}.crm.dynamics.com/main.aspx>

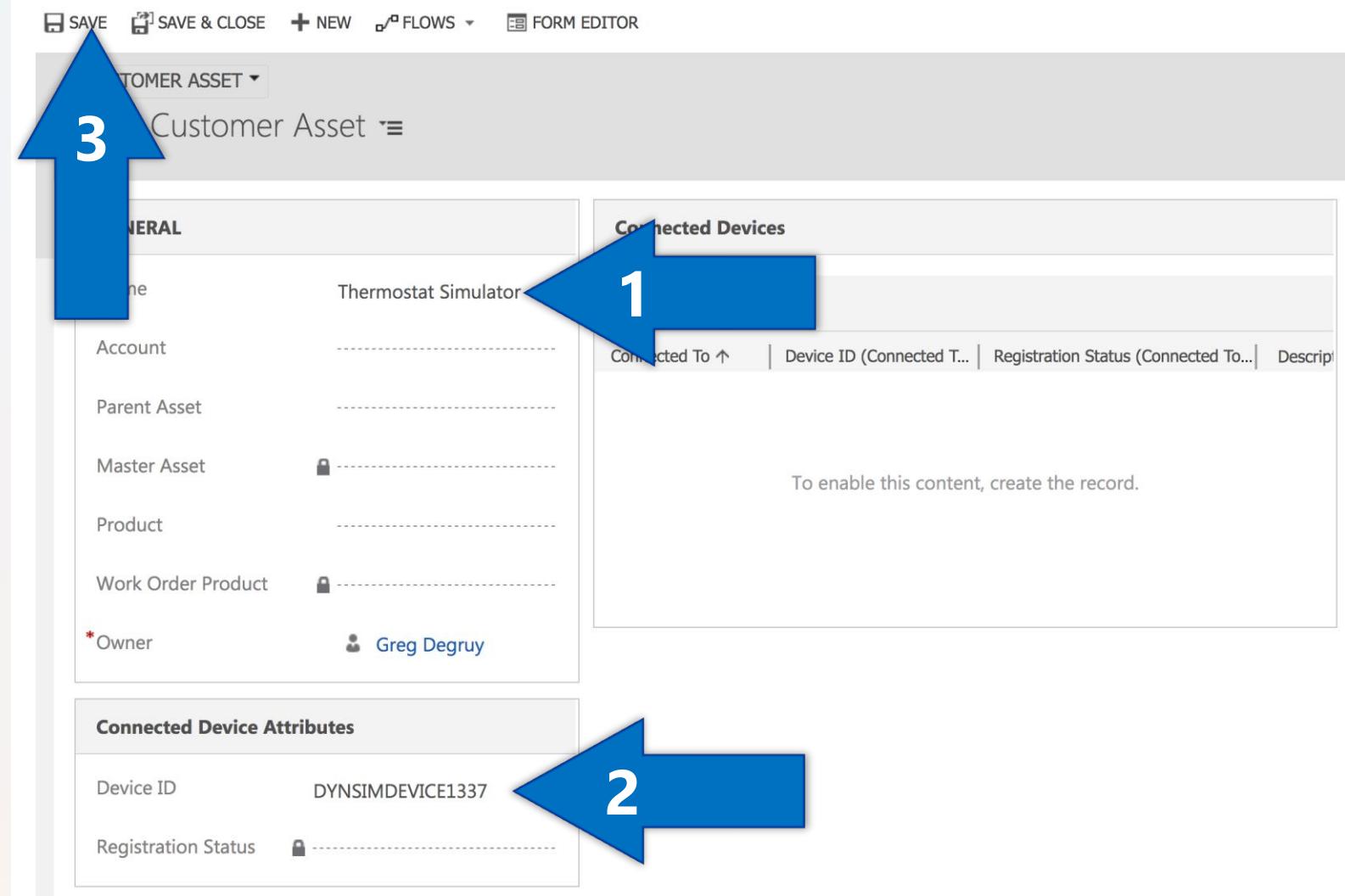
- `1` On the navigation bar click on the drop down next to Sales
- `2` The Sitemap will appear, click on Field Service
- `3` Click Customer Assets.



New Customer Asset

Make sure this Device ID is unique from any existing Device IDs in IoT Hub.

- '1' Add an asset name, in my case Thermostat Simulator
- '2' Add a unique Device ID that does not exists in your IoT Hub, in my case I used DYNSSIMDEVICE1337
- '3' Click Save



Register with IoT Hub

For a moment you'll see "The asset is being registered in the background". You can continue to use Microsoft Dynamics CRM. Takes seconds to complete.

CFS LAB 1

The screenshot shows the Microsoft Dynamics CRM interface for managing Customer Assets. At the top, there's a toolbar with actions: NEW, DEACTIVATE, DELETE, REGISTER DEVICES, CREATE COMMAND, ASSIGN, SHARE, EMAIL A LINK, RUN WORKFLOW, and more. Below the toolbar, the title bar says 'CUSTOMER ASSET' and the specific asset name is 'Thermostat Simulator'. The main area is divided into sections: 'Summary', 'GENERAL', 'Connected Devices', and 'Connected Device Attributes'. In the 'GENERAL' section, fields include Name (Thermostat Simulator), Account, Parent Asset, Master Asset (locked), Product, Work Order Product (locked), and Owner (Greg Degrugy). The 'Connected Devices' section shows a table header with columns for Connected To, Device ID (Connected To...), Registration Status (Connected To...), and Description. Below the table, it says 'No Connections found for this Customer Asset. Select Add (+)'. In the 'Connected Device Attributes' section, the Device ID is listed as DYNSSIMDEVICE1337 and the Registration Status is shown as locked.

+ NEW DEACTIVATE DELETE REGISTER DEVICES CREATE COMMAND ASSIGN SHARE EMAIL A LINK RUN WORKFLOW ...

CUSTOMER ASSET

Thermostat Simulator

Summary

GENERAL

*Name: Thermostat Simulator

Account: [redacted]

Parent Asset: [redacted]

Master Asset: [locked]

Product: [redacted]

Work Order Product: [locked]

*Owner: Greg Degrugy

Connected Devices

Connected To ↑ | Device ID (Connected To...) | Registration Status (Connected To...) | Description

No Connections found for this Customer Asset. Select Add (+).

Connected Device Attributes

Device ID: DYNSSIMDEVICE1337

Registration Status: [locked]

Connected Device

This makes the Customer Asset a connected device giving us more advanced monitoring capabilities and access to more data points in Dynamics with the device properly tied to IoT Hub.

The screenshot shows the Microsoft Dynamics 365 interface for a Customer Asset named "Thermostat Simulator".

General Information:

- Name: Thermostat Simulator
- Account: (Blank)
- Parent Asset: (Blank)
- Master Asset: (Locked)
- Product: (Blank)
- Work Order Product: (Locked)
- Owner: Greg Degruy

Connected Devices:

Connected To ↑	Device ID (Connected To...)	Registration Status (Connected To...)	Description
Thermostat S...	DYNSIMDEVICE1337	Registered	(Blank)

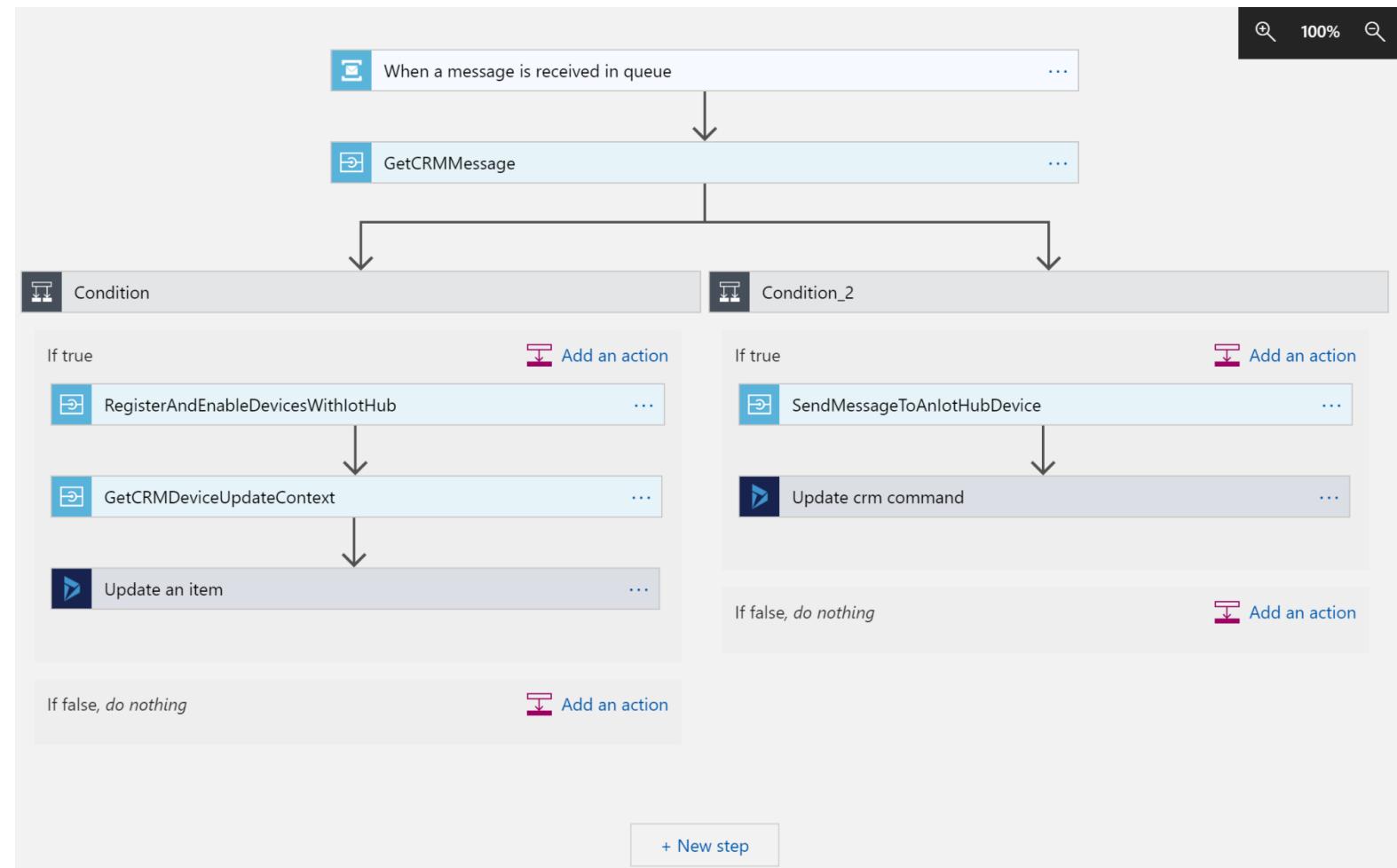
Connected Device Attributes:

Device ID	DYNSIMDEVICE1337
Registration Status	Registered

CRM-To-IoT Logic App

The CRM-To-IoT Logic App handles the heavy lifting for

1. Capturing the create event for this new device
2. Queueing the message
3. Identifying it as a device create message
4. Taking the necessary steps to register and enable the device(s) with the IoT Hub sending the completed registration message back to Dynamics



Thermostat Simulator

Now the new Dynamics connected device accessible in our Thermostat Simulator.

Got back to the Thermostat Simulator web app and select the newly registered device (*you will need to refresh the page*).

Optionally, if you visit your IoT Hub you'll see the device information present there as well.

The screenshot shows a web-based interface for managing a connected device. At the top, there's a navigation bar with tabs: 'Field Service IoT' (selected), 'Thermometer', 'Connection', and 'Reboot'. To the right of the tabs, it says 'Connection Status: Connected'. Below the navigation is a large image of a digital wall-mounted thermostat with a blue display showing '25.3 °C' and '18.0 °C'. A hand is pointing at the device. To the right of the image is a control panel. It includes a 'Device ID' dropdown menu with options: 'Select a device', 'AZ3166', 'DYNSSIMDEVICE1337', and 'SIMDEVICE1337'. A blue arrow points upwards from this menu towards the 'Temperature' gauge. The 'Temperature' gauge is green and shows '65°F'. The 'Humidity' gauge is blue and shows '40%'. Below the gauges, there are sections for 'Messages Received:' and 'Messages Sent:', each with a long green progress bar.

Field Service IoT Thermometer Connection Reboot

Connection Status: Connected

Device ID

Select a device
AZ3166
DYNSSIMDEVICE1337
SIMDEVICE1337

Refresh

Humidity 40%

Temperature 65°F

Messages Received:

Messages Sent:

Threshold Temperature

Turn the temperature past the 70 degree threshold again if it's not already.

Field Service IoT Thermometer Connection Reboot

Connection Status: Connected

Device ID: DYN5IMDEVICE133 Refresh

Humidity: 40%

Temperature: 83°F

Messages Received:

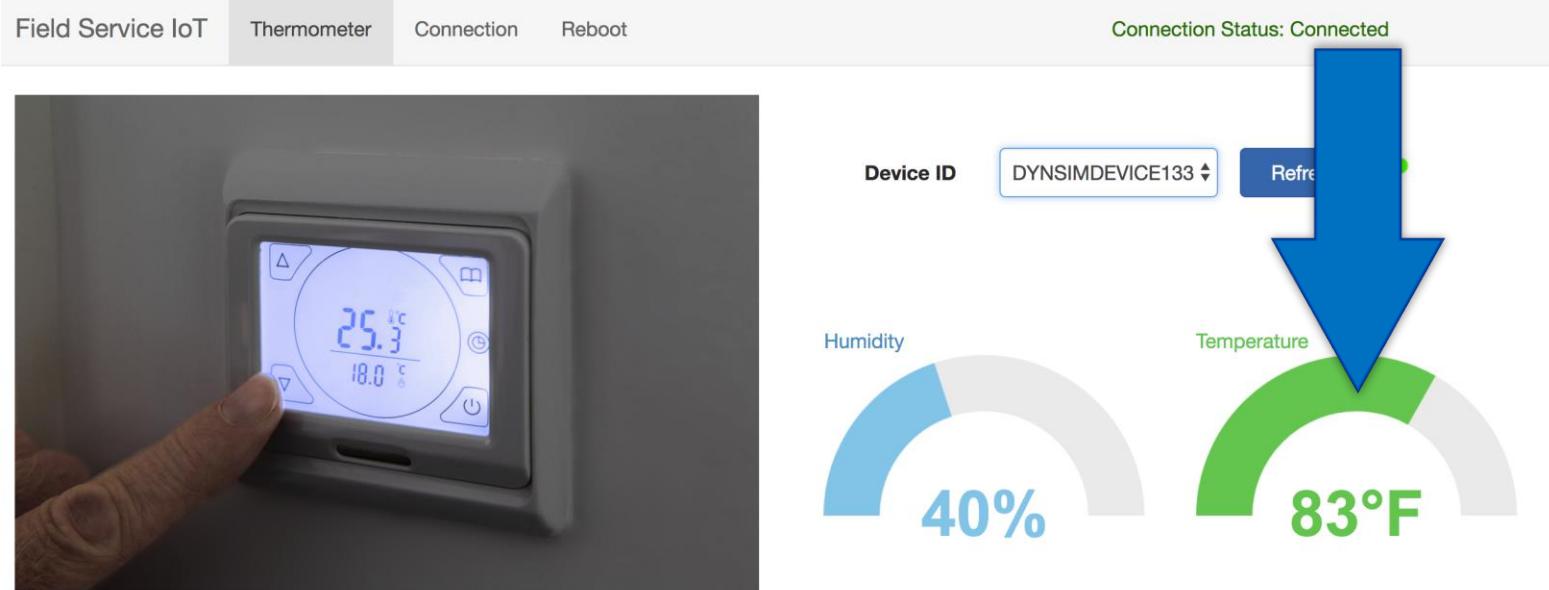
CURRENT STATUS --> Temperature: 83 Humi

CURRENT STATUS --> Temperature: 83 Humidity: 40

Messages Sent:

CURRENT STATUS --> Temperature: 83 Humi

CURRENT STATUS --> Temperature: 83 Humidity: 40

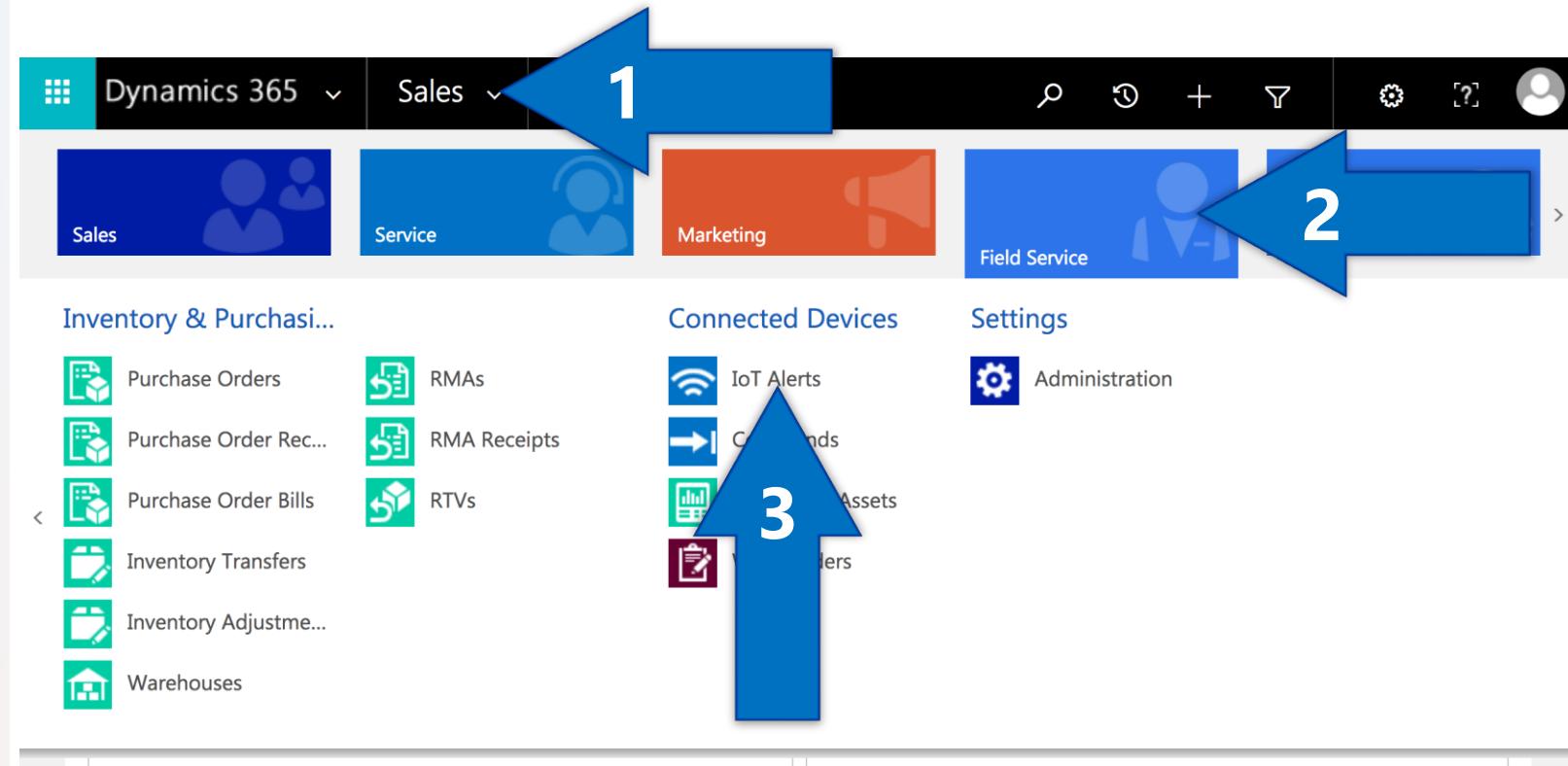


Dynamics 365

In your browser go back to Dynamics 365 at your organization URL or just click on the tabs it's in if you haven't closed it.

The URL should look something like this
<https://{{YOUR ORGANIZATION NAME HERE}}.crm.dynamics.com/main.aspx>

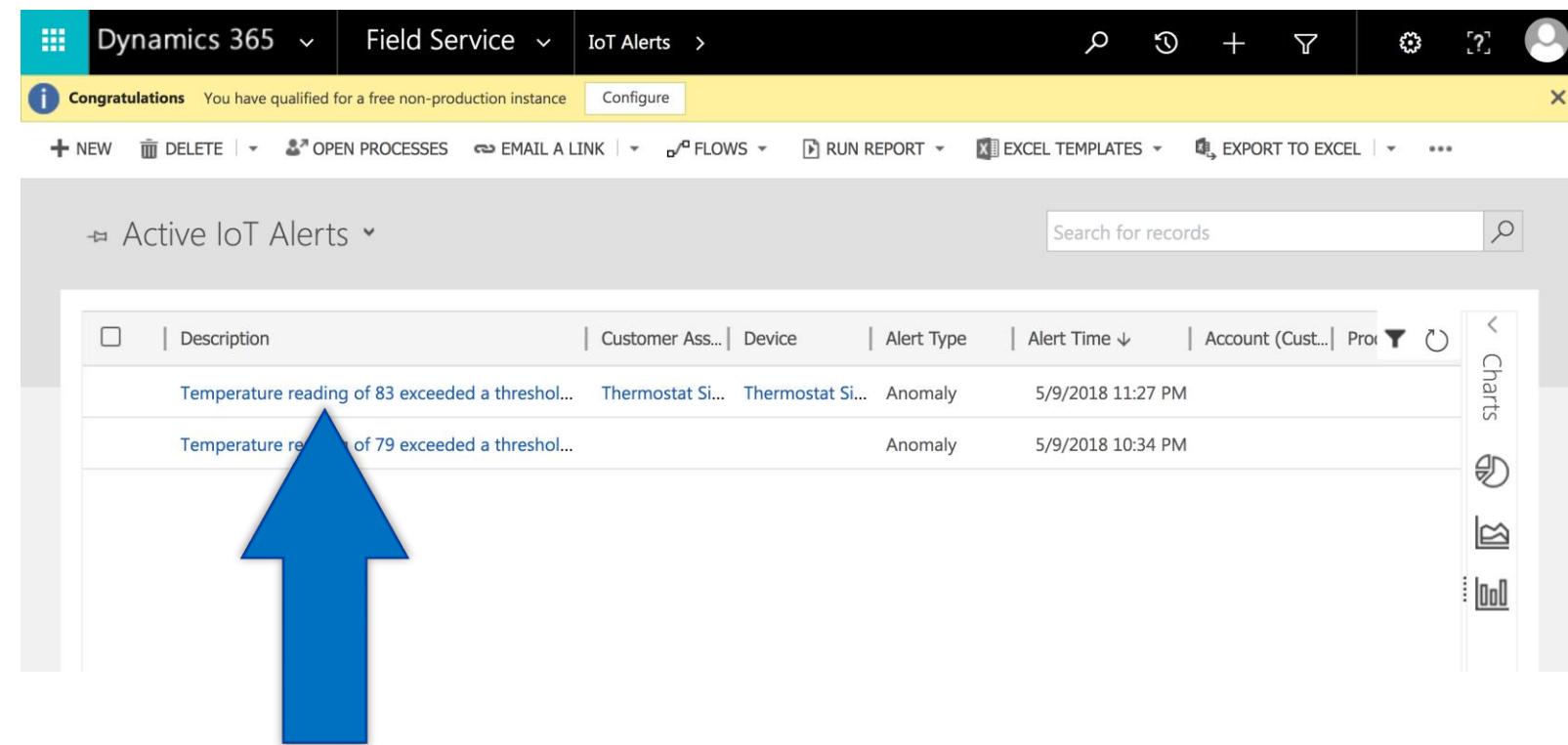
- `1` On the navigation bar click on the drop down next to Sales.
- `2` The Sitemap will appear, click Field Service.
- `3` Click on IoT Alerts.



Basic IoT Alert

Nice! A new complete Active IoT Alert has been captured.

Optionally, if you visit the IoT-To-CRM logic you'll also notice a new Succeeded Run has been completed. If you click it into you can see run details step by step in the logic app as well as the data piped through each step.



A screenshot of the Dynamics 365 Field Service IoT Alerts page. The page title is "Active IoT Alerts". There are two rows of alert data:

Description	Customer Ass...	Device	Alert Type	Alert Time	Account (Cust...)	Pro
Temperature reading of 83 exceeded a threshold	Thermostat Si...	Thermostat Si...	Anomaly	5/9/2018 11:27 PM		
Temperature reading of 79 exceeded a threshold			Anomaly	5/9/2018 10:34 PM		

A large blue arrow points upwards from the bottom of the slide towards the screenshot.

Complete IoT Alert Information

Now this new IoT Alert Entry has the complete Customer Asset information we expect and is ready to be filed as a case, schedule and dispatch of a technician to respond to the Anomaly.

IOT ALERT : INFORMATION

Temperature reading of 83 exceeded a threshold of 70

* Owner
Greg Degruy

Created (Active for 5 minutes) ➔ **Create Case** ➔ **Create Work Order** ➔ **Schedule Work Order** ➔ **Close Work Order**

✓ Device Thermostat Simulator
 ✓ *Description Temperature reading of
 ✓ Alert Time 5/9/2018 11:27 PM

CFS - IoT Alert Process Flow (Active for 5 minutes) [Next Stage](#) [^](#)

General

*Description	Temperature reading of 83 exceeded a threshold of 70
Alert Type	Anomaly
Alert Token	342decb1-1750-4715-b9c8-6915b2fdc3b3
Alert Time	5/9/2018 11:27 PM
Alert URL	-----
Alert Data	{"deviceid":"DYNSSIMDEVICE1337","readingtype":"Temperature","reading":83.0,"eventtoken":"342decb1-1750-4715-b9c8-6915b2fdc3b3","threshold":70.0,"ruleoutput":"AlarmTemp","time":"2018-05-10T06:27"}

ASSISTANT **NOTES**

Enter a note

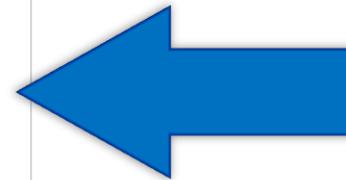
There aren't any notes to show. To get started, enter a note.

Connected Device Readings

Add a Power BI tile for the connected device.

CUSTOMER ASSET

Customer Asset	Thermostat Simulator
Device	Thermostat Simulator
Device ID	DYNSSIMDEVICE1337



Completed Basic CFS Deployment



In the coming labs we'll begin to walk through how to work with real IoT Devices with Connected Field Service

To Save Cost Stop the following services if you do not plan to continue the other labs.



Thank you for your time in learning more about CFS and completing this tutorial.



API App Services

- Queue Message Parser
- IoT Hub
- CRM Helper



Web App Service

- Thermostat Simulator