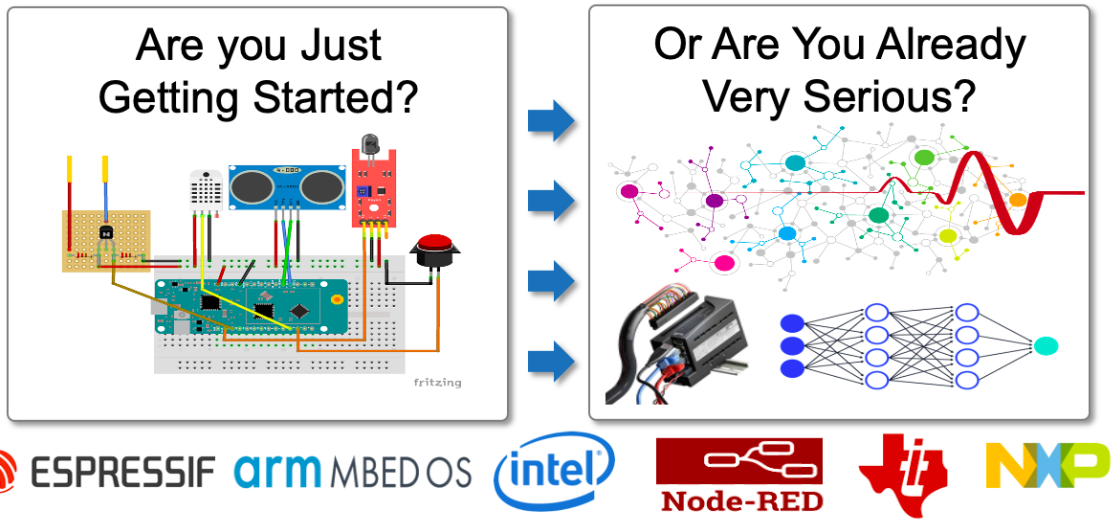
“What are you working on?”



Flex83

IoT Application Builder

Onboarding Guide

Revision 2.0

IoT83 Ltd.

450 South Abel Street #361625

Milpitas, CA 95036, USA

Phone: +1 650 455 6961

https://iot83.com/

Table of Contents

1 Introduction

This document presents the steps, procedures and guidelines for building IoT applications using IoT83's "Flex Version" of its Application Enablement Platform, Flex83.

This Flex83 product provides a powerful and cost effective path to create dynamic IoT Applications. Should any users of this platform desire to move up to the full power of the Enterprise Versions of Flex83, please contact IoT83 for details (https://iot83.com/contact-us).

1.1 About

The **Flex83 Portal** is a cloud-based platform that is offered to users in SaaS offering depending to the scale and use-cases needed for your application. Using the platform tool sets, users can easily onboard any IoT Device to the platform, start visualizing device data, apply rules and policies to the streaming data and set alarms or alerts in form of emails and SMS or "Action on Events" in the form of Remote Procedure Calls (RPCs) or Webhooks. Flex83 cloud offers Secure MQTT connectivity from the End user's device to the cloud.

One of the unique aspects of Flex83 is that as a user configures their subscription by onboarding devices, device data, setting up device groups, device data attributes, etc., the platform is dynamically building a professional grade application in the background. Once a user's devices are connected and configured, the foundation of a powerful user application is already created and available. Further, Method83 has debugging tools integrated at every layer of the platform so that you always can validate your progress and quickly see results.

1.2 Acronyms Used

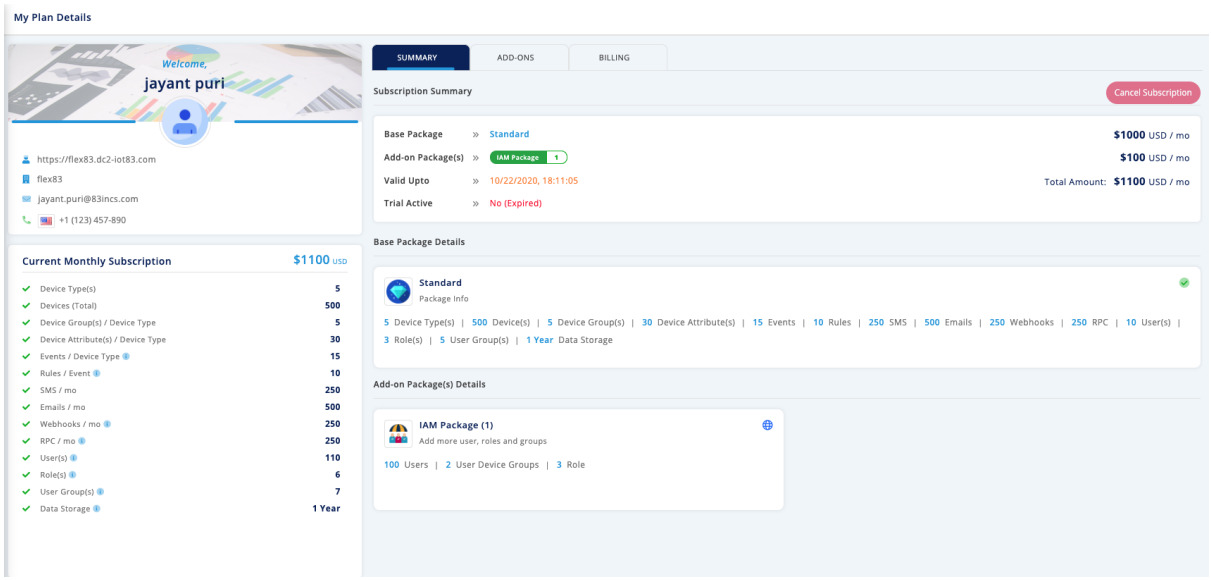
Term	Description
F83	Refers to the Flex83 Cloud
IOT	Internet of Things
MQTT	Message Queuing Telemetry Transport

1.3 Revision History

Rev	Date of Issue	Author	Scope
0.1	2020-06-03	IoT83 Ltd	Initial Draft
0.2	2020-06-21	IoT83 Ltd	Updated Usage Guide
1.0	2020-09-04	IoT83 Ltd	Updated for Product Release

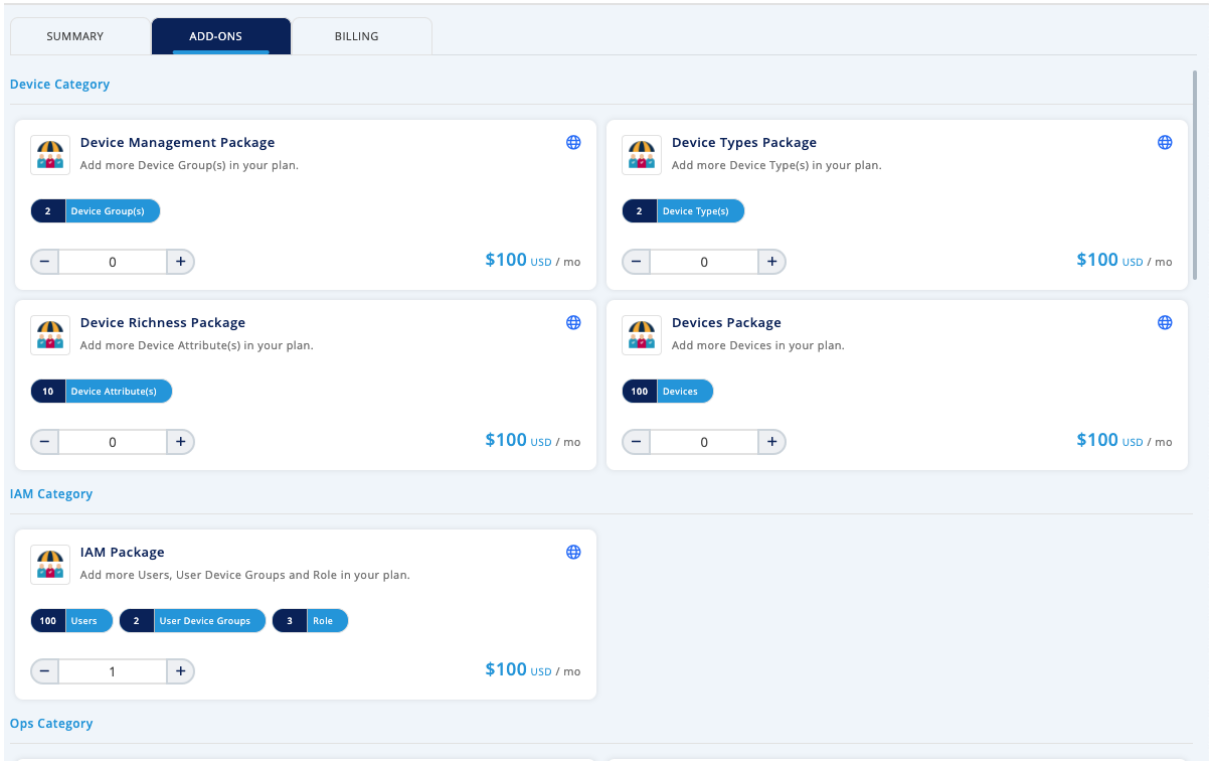
2 My Account

The landing page of the portal is the “My Account” page, showing your current **plan details** and various options available regarding your subscription plans along with options to upgrade to a higher plan, cancel the current plan, check billing details, payment history and more.



2.1 Subscription Plan

Flex83 offers a base plan starting at $1000/month. Post this it offers add on services/quota that can be purchased if you want more resources. Paid plan is available by signing up for the plan’s monthly recurring amount using **Paypal** as the payment gateway provider. On this page you can always see your current subscription and the usage entitlements enabled for that plan.



2.2 Billing Details

This section shows all the transaction history for your account. This includes, purchases, cancellations and refunds, if any. It also provides a section where you can also check your invoices for all the payments made.

2.2.1 Information on Plan Details:

1. Trials are valid for 45 days (subject to change by IoT83 at any time) and once in a lifetime for an email address / phone number.
2. After Expiry of a Trial your card will be automatically charged and your account will get converted to a paid subscription.
3. Cancelling your subscription for the month for which the billing is in process (cancellation date same as billing date) will not result in a refund for the balance of the month.
4. Payments are charged at the beginning of each billing cycle.

2.3 Plans

This section details the options available under each plans and addons, beginning with an overview of the building blocks that make up each plan. By understanding how many of each of these plan building blocks, you can understand which plan will be needed for the use-cases, application, or applications you wish to build and run.

a) Number of Device Types – Define the unique device types that you are entitled to. A "Device Type" is defined as any device that provides the same JSON format when it connects to Flex83 over MQTT. So, from the point of view of Flex83, a Raspberry Pi and an Arduino can be the same Device Type, if they are sending the same JSON data to the platform.

b) Number of Devices – This denotes the actual end-user devices, or maximum allowed unique MQTT connections to the Cloud. The Devices allowed are shared across Device Types, meaning, if you are allocated 4 Device Types and 20 Devices, you can use a total of 20 Devices across all these 4 Device Types.

c) Number of Device Groups – Groups are administrative settings to allow users to club together different devices under one logical namespace adding a lot of value in managing device events.

d) Number of Device Attributes – Device Attributes are the "Name / Value Pairs" that your devices send to the Platform in JSON format. The number of device attributes sets the size of the JSON that your devices can forward.

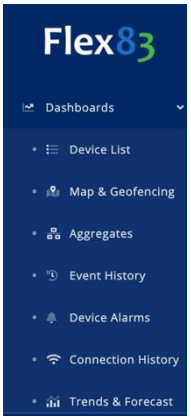
e) Number of Monitoring Events – In Flex83 "Events" are basically combinational policies that watch for a certain conditions that you program that on detection of the event trigger the "Event". Events can trigger emails, SMS, Remote Procedure Calls (RPCs), or Webhooks, and are a very powerful platform feature.

f) **Number of Event Rules –** The logical statements that make up the Events are called rules.
g) **SMS / Emails –** Based on your plan, you are entitled a particular number of SMS and Emails that you can ask the system to send on your behalf. These are primarily used for Policy and rule alerts or alarms, so that you are able to receive notifications in form of email and SMS for your devices based on the rules that you configure. Should you run out of this resource, the system will notify you.
h) **Webhooks -** The HTTP based calls that can be triggered as actions to events.
i) **Remote Procedure Calls -** The control call that can be made to the device via the portal.
j) **Data Retention –** The cloud stores the raw data for a specified duration determined by the plan you choose. Any data that becomes stale or falls outside the criteria is cleaned up time to time.

3 Understanding the Flex83 Left Menu

Below are the navigation options for the 5 menus of Flex83 Platform (minor changes to the arrangement of these menus may occur from time to time, but the basic logic of the implementation will remain the same).

Each main menu item , or Parent, represents a major platform function or logical section. The “child” items in each section provide access to the logical sub-topics of that application section, as shown in to the left and listed below:



3.1 Dashboards

3.1.1 Device List

3.1.2 Maps and Geofencing

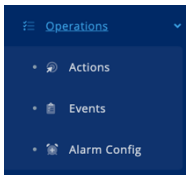
3.1.3 Aggregates

3.1.4 Event History

3.1.5 Device Alarms

3.1.6 Connection History

3.1.7 Trends & Forecast

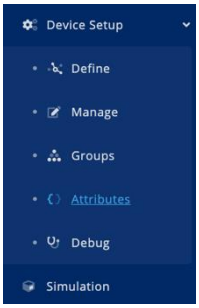


3.2 Operations

3.2.1 Actions

3.2.2 Events

3.2.3 Alarms



3.3 Device Setup

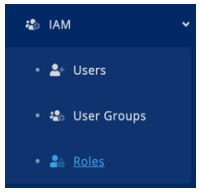
3.3.1 Define

3.3.2 Manage

3.3.3 Groups

3.3.4 Attributes

3.3.5 Debug



3.4 IAM

3.4.1 Users

3.4.2 User Groups

3.4.3 Roles

3.5 Simulation

3.6 Help

Again, One of the unique aspects of Flex83 is that as a user configures their subscription by onboarding devices, device data, setting up device groups, device data attributes, alarm profiles, and event monitors, etc., the platform is dynamically building a professional grade application in the background. The rest of the Document will show how this works and cover how easy it is to configure the system and create your powerful application.

4 Device Setup

The Device Setup process is very streamlined and once you are familiar with the process, you can skip steps come back to them later to get up and running quickly. But to understand the full power of Flex83 the step-by-step process we will follow here leverages all the Device Setup and also strongly recommend using the simulation functionality So before we start, here is the outline of what we will walk through in the following sections:

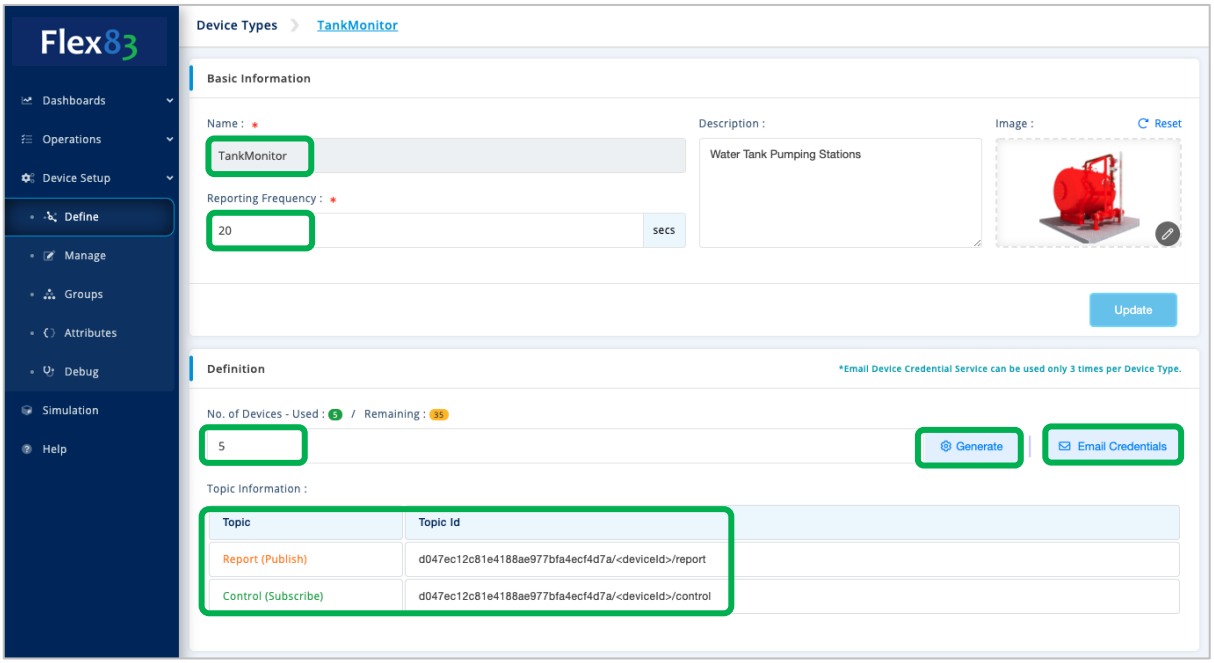
1. First we will define a Device Type following the prompts on the Define Page.
2. Then, we will follow the prompts on the Manage Page to name the individual devices and establish their locations.
3. Next, we will show the features of the Groups & Tags Pages. Here I like to create two groups Simulated and Physical Devices (more on this later).
4. Next, you can jump to creating a simulation (targeting the Simulated Device Group) to push some data to our application and create the Schema for the Device Type we created.
5. Next, we will open the Attributes Page to “decorate” our data, and to possibly create some “synthetic” or “derived” variables or attributes.
6. Next, we will look at multiple examples of how to connect “Physical Devices” to Flex83 and get this working several ways.
7. Finally, we will validate that both our physical and simulated devices are connected as intended and then tour our dashboards to see how Method83 has dynamically created our application.

So let’s get started!

4.1 Define

This is the step where you define your Device Type. For example, if you are trying to connect a Device Type called *TankMonitor*, then you would name your device as shown below, indicate how often you want the device to send updated data, add an image to represent the Asset Type (which will automatically be leveraged in your application dashboard pages) and click “Update”. Once this step is completed, you can identify how many Devices of this Device Type you wish to create, and then click “Generate” to create the unique Flex83 identifiers of the devices.

At this time, you can also click “Email Credentials” to send a file to your registered email account that contains all the device connectivity information and security / authentication credentials needed to connect your devices the Device Type instance you just created. We will examine this file later when we connect our devices, but it is IMPORTANT TO NOTE that each Device Type has unique Credentials. That is, if you have 3 Device Types, you will have 3 Credentials files uniquely identifying each.



As circled in green above, the system will assign you a pair for Topics for a two way Connection from *Device to Cloud* and *Cloud to Device*, for Report and Control topics respectively.

The top Topic is your device’s MQTT publish topic, where your device will publish to the unique Flex83 identifier for your Device Type in the format:

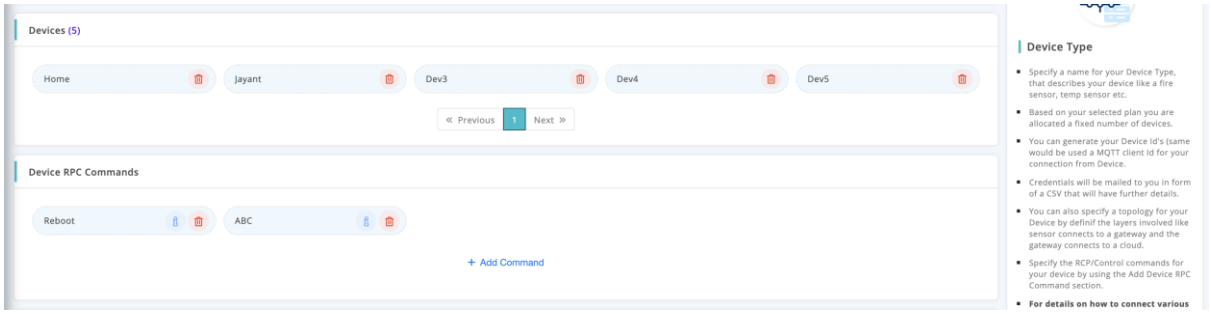
<unique_device_type_identifier_string>/<unique_device_identifier_string>/report

Similarly, you device can subscribe to MQTT messages from Flex83 to you device using the format:

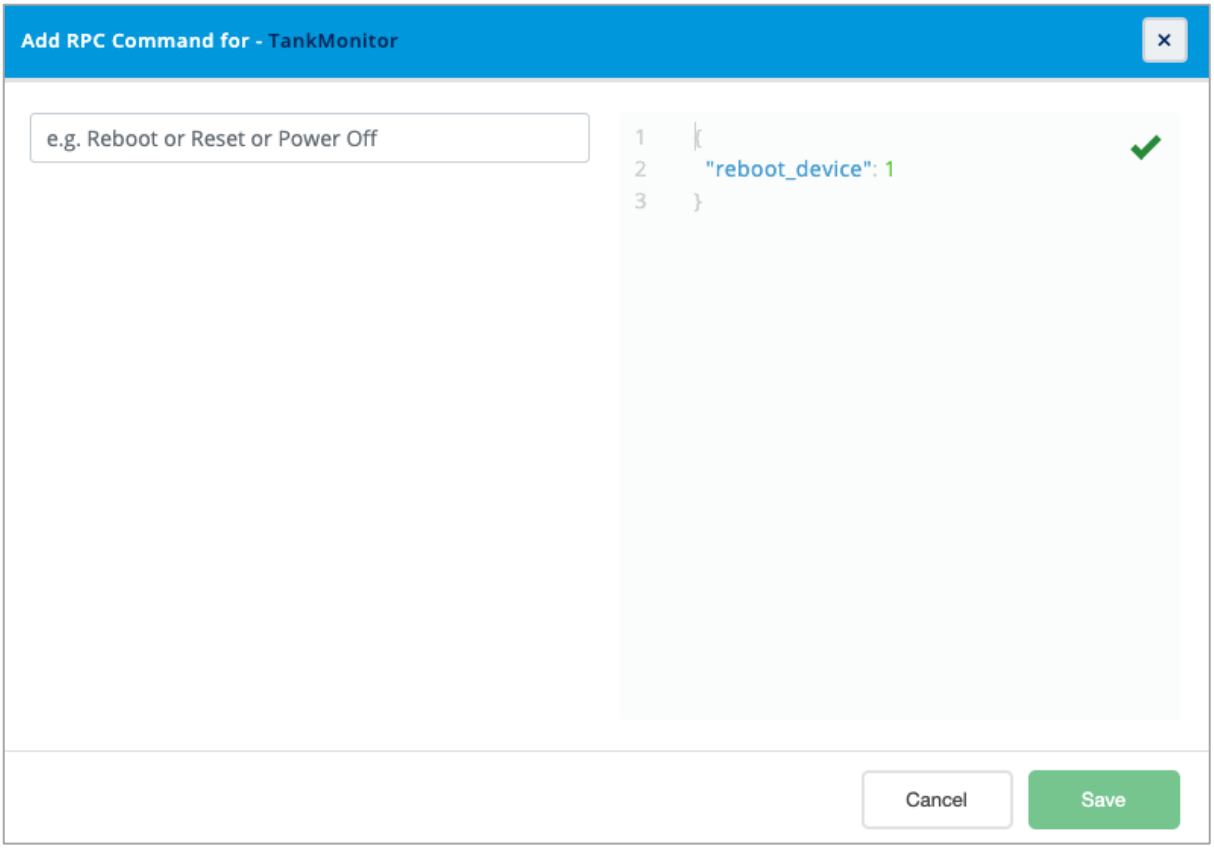
<unique_device_type_identifier_string>/<unique_device_identifier_string>/control

Note that the “unique_device_identifier_string” is found on the “Manage” page of Device Setup.

Further down on the “Define” page, you can see your actual devices (which you will name in the next step). But below that a section where you can add remote procedure calls to your devices.

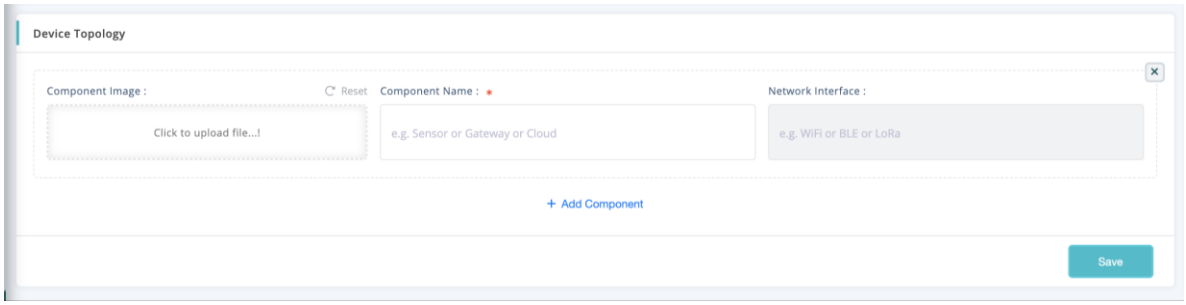


You may choose to come back to this option later, but when you click “Add Command” a form like this will pop up”:

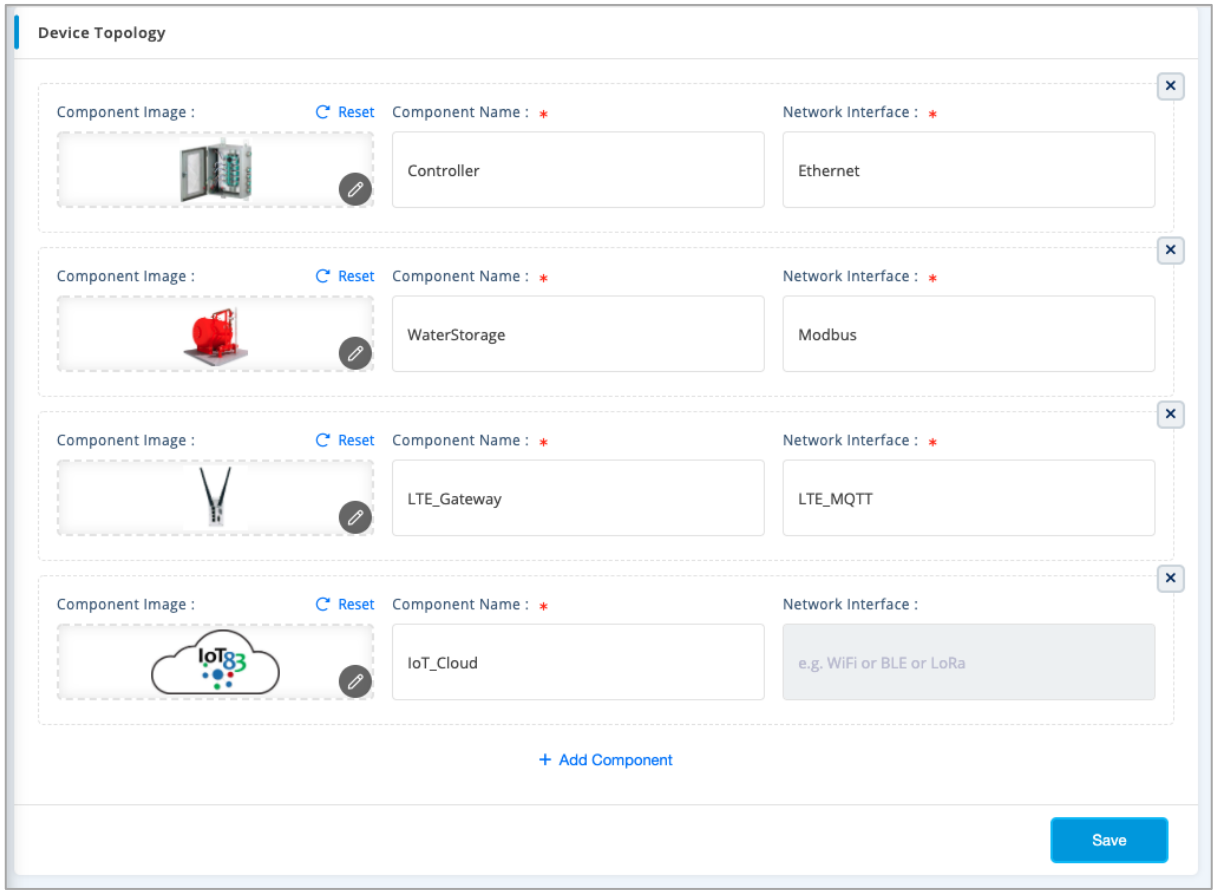


Here, we are imagining that under certain circumstances we may wish to reboot our device, and so we create an RPC called Reboot. The contents of this Reboot RPC are simply a JSON that will be passed to the device using the “control” MQTT channel we just discussed. This RPC can then be the “Event” triggered in an Event policy. For example, one of your Event policies may be watching for circumstances indicating a fault in your device, where upon detection, you would wish to reboot the device. This RPC, then could be triggered and sent to your device, then interpreted by the device, triggering the reboot.

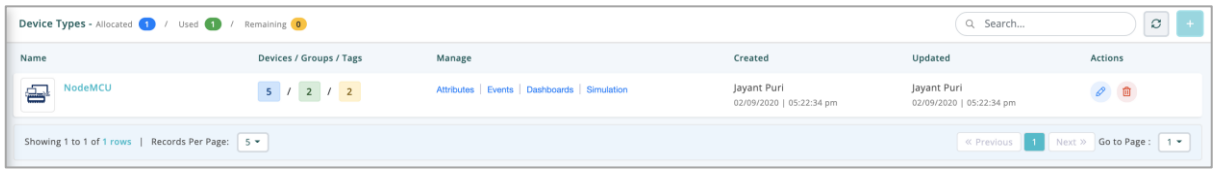
Further down the page, you will see a section (which is optional, but very nice when completed) where you can add icons indicating your device deployment and connectivity. You may wish to fill this in later as well, but let’s go ahead and clarify this here.



Here is an example implementation showing a “chain of connectivity” to the end device which is a water pumping station. The Station is connected to a controller, which is connected to a gateway, which is connected to the cloud. So, we set up this page like so to be able to show this connectivity in our application:



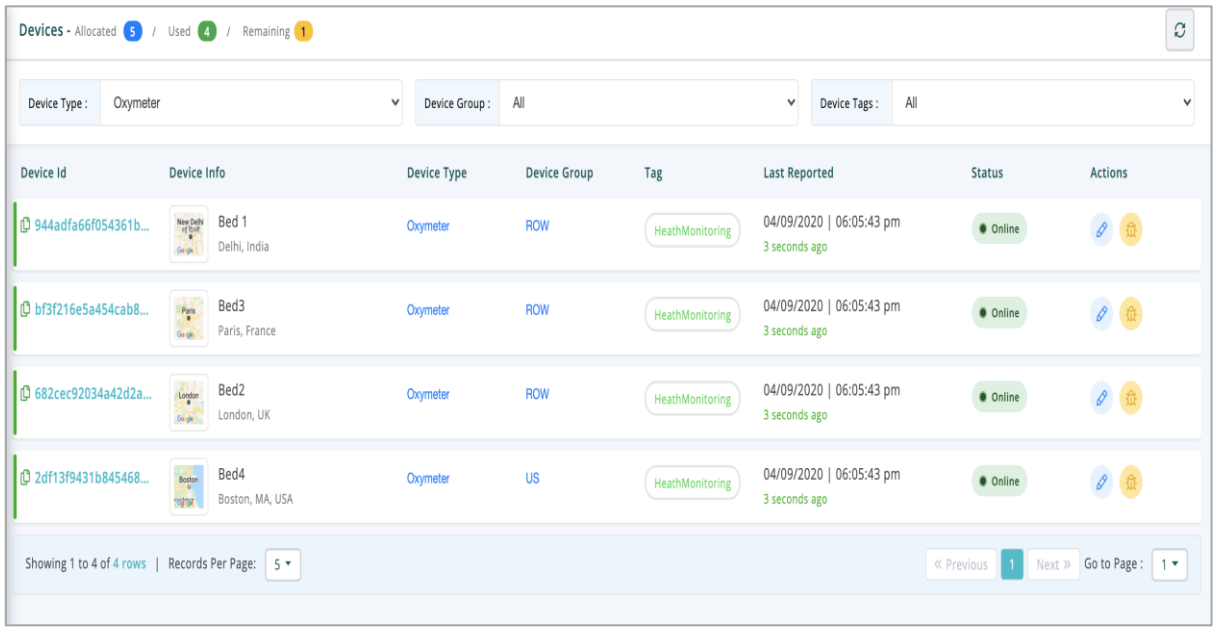
That completes the “Define” page set up. Once this is all completed and saved, at the device list screen, you should see the instance of the Device Type you just created. You can click the edit icon on the right to re-visit configuration (perhaps to change you Device Type image), or delete the Device Type if for some reason you need to.



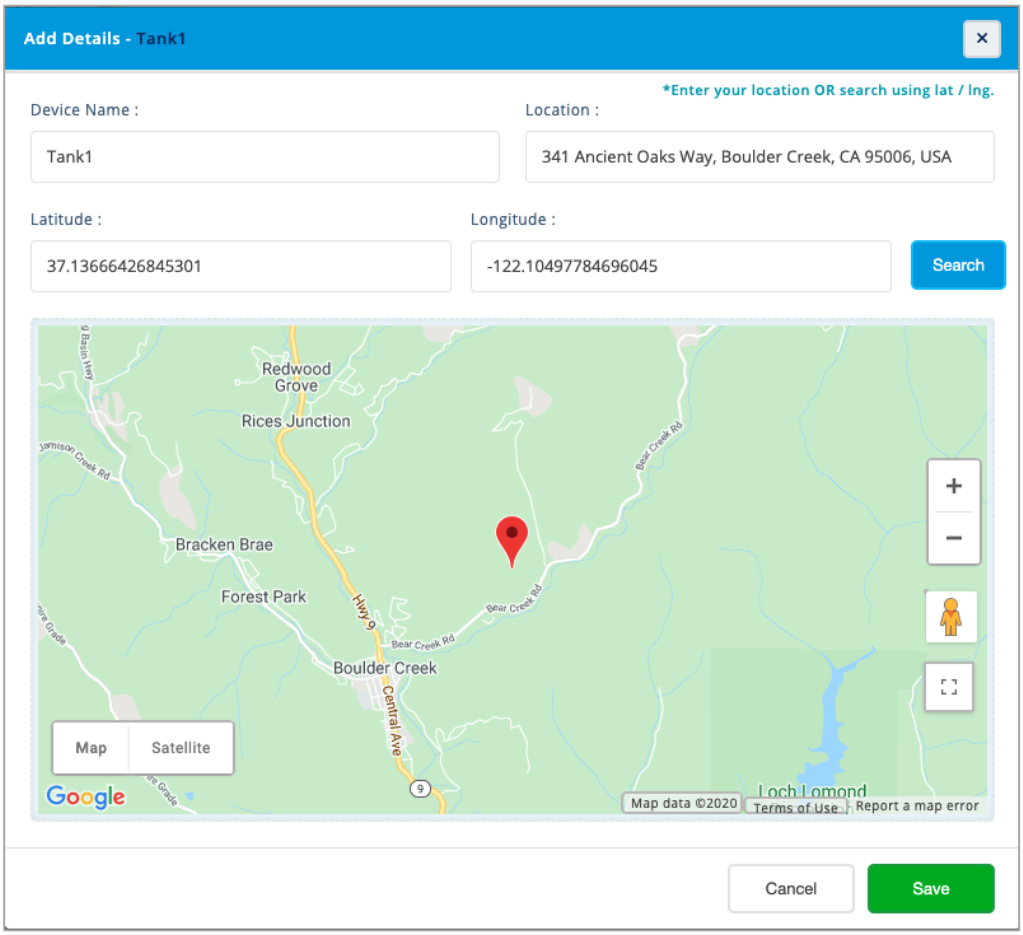
Note that the plan you have selected determines the number of Device Types and the total number of Devices that you can create using this Connector service.

4.2 Manage

In this Section, we show how to configure your Device names and locations, a very quick & easy process. First, in the Device Setup menu area, select “Manage”, and you should see a screen pop up that looks like the following:

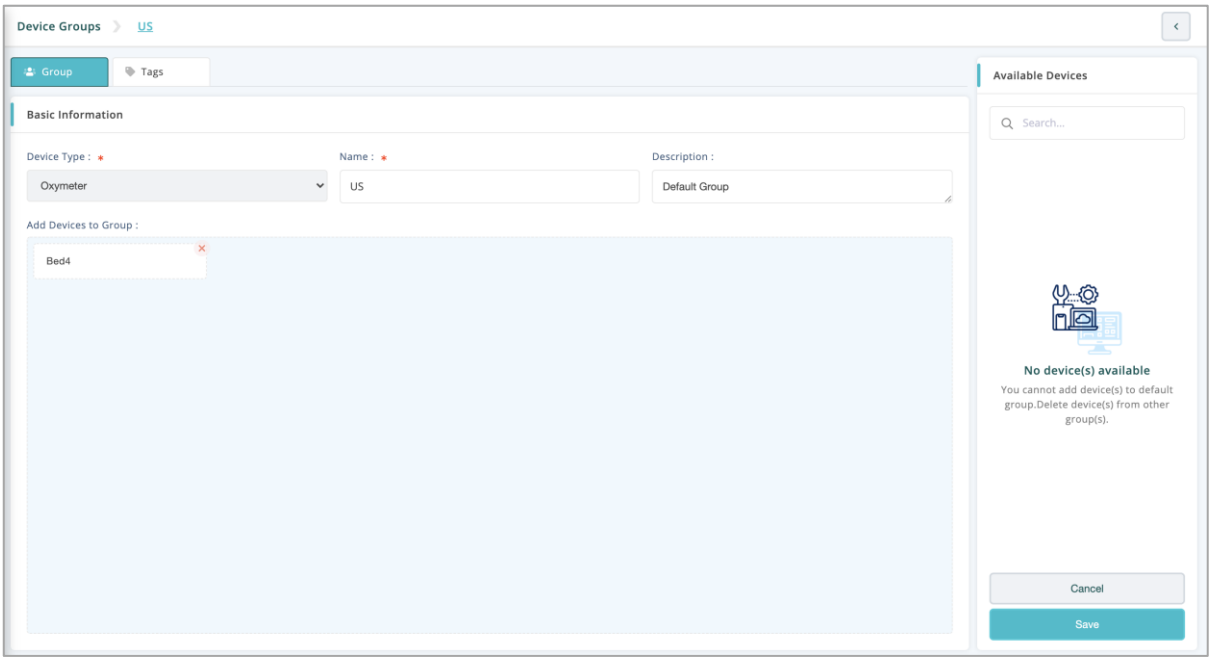


You can select the edit icon on the right for each of your devices to name them and set their locations. When you do, a screen that looks like the following will pop up. You can name your device and then zoom in to any geography and add in the address of the device, and save it (for each one). Once placed, you can also drag the location and base the location of Latitude and Longitude, or place the device directly using lat / long from the start.

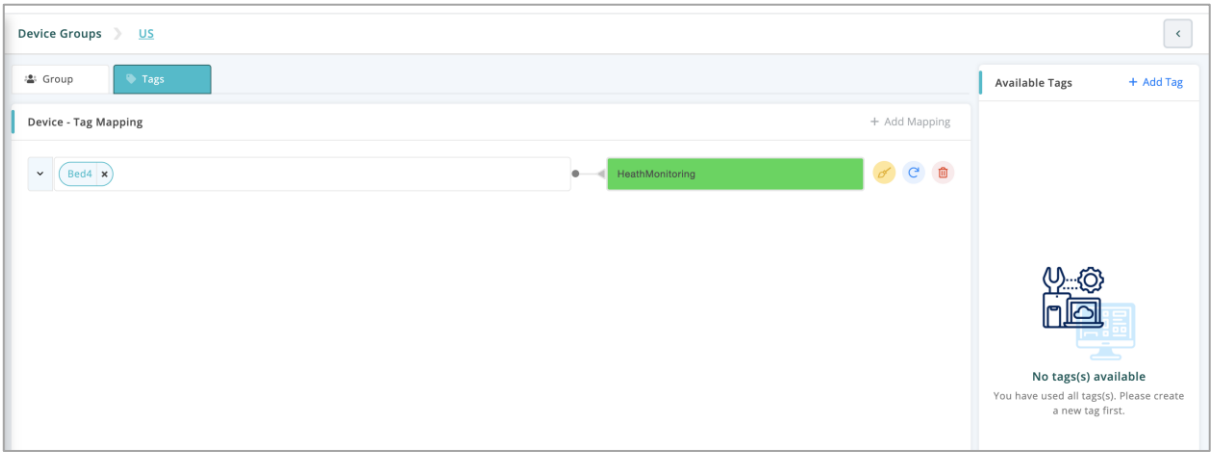


4.3 Groups

This section allows to perform administrative functions on your Devices like – Classifying them into groups or tagging them etc. So, under Device Setup, click Groups, and Create New. This will bring up the Create Group screen where you can name the group, set its description, and add devices to the group. Once complete, add other groups to organize your devices into the most logical organization for your application.



Next click Tags. Here you can create tags for your Devices. Perhaps you want to tag by criticality, priority, function, or some other indicator. In practice you can use Tags to sort for different Devices that you want to monitor, analyse or watch more closely. To begin, simply click “Add Tag” on the right, select a color for the tag, name the tag and it becomes available as a tag. Once you have created the tags you want, connect them to Devices according to whatever scheme you have identified for tagging devices in your application.

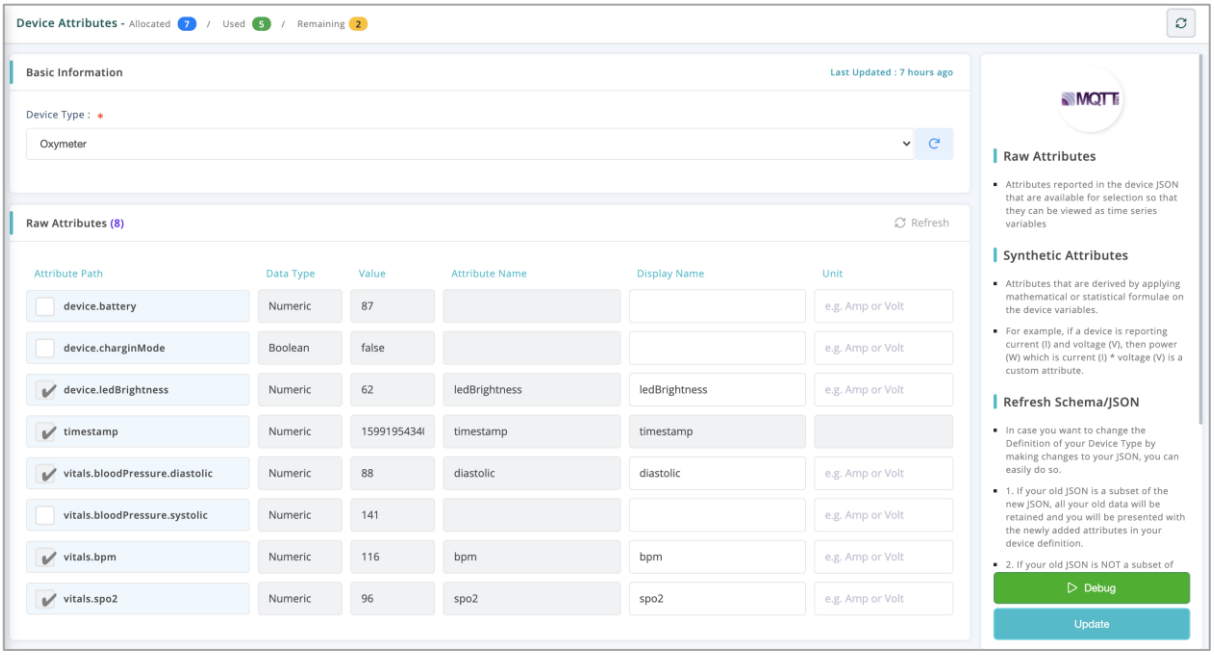


Note: Default group can only be renamed and cannot be deleted. You can add Devices to a default group but cannot take them out. In case you want to create another group, then you will need to add devices to this group and they will automatically move out of the default group. A device can sit in **only one group**. By Default every device that gets created is sent to the default group

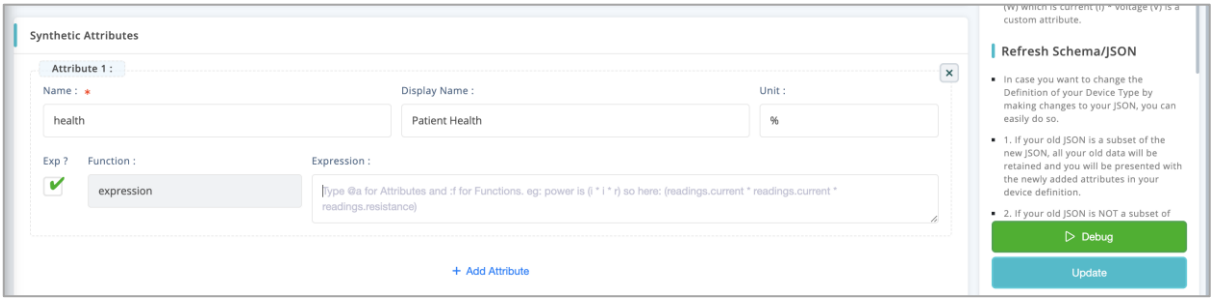
4.4 Attributes

Once you have either simulations running (see below) or have connected your devices to the cloud (also below), the data attributes the devices are forwarding (that is the elements in the MQTT JSON being sent) will automatically appear in this section (Device Setup / Attributes).

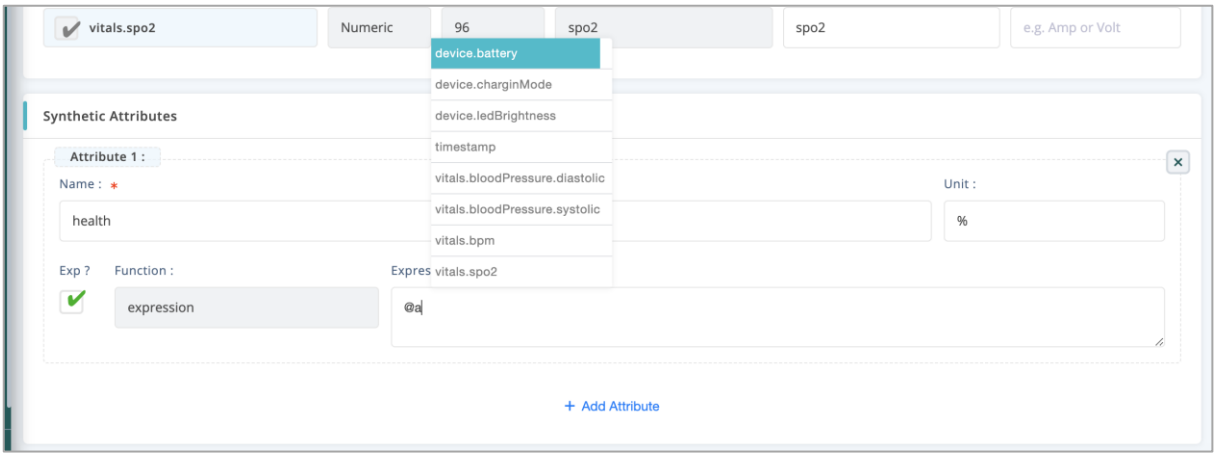
This is one of the most powerful sections as you can do a lot here. Not only can you select your RAW attributes for integration into your application but you can also create custom / synthetic attributes as shown below. Once you create your custom attributes, you can also debug to see the value of the attribute



In the screen above, some of the Device attributes have been selected to include in the application dashboards, and assigned friendly names, and perhaps units have been assigned to the attributes.

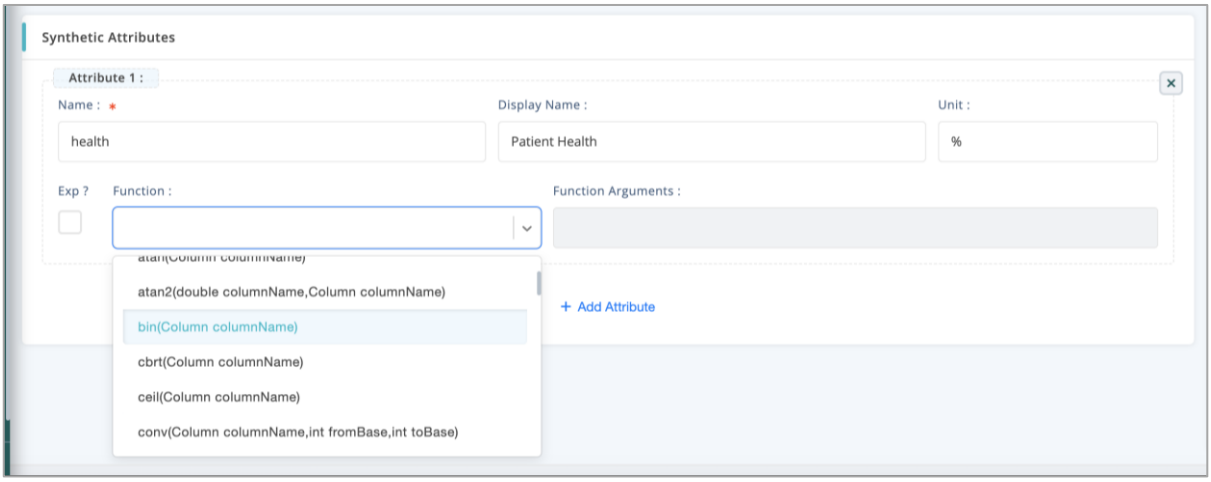


In this section of the Attributes page, you can create “custom” or “synthetic” attributes. A synthetic attribute is one that is created using some logical expression using a combination of attributes, such as creating a power attribute by multiplying current and voltage, or by otherwise applying some math function on a single attribute. The possibilities here are quite extensive. To better understand that take a look at the following screen shot:



In the Expression box, but typing “@a”, a dialogue box appears where you can select any of your Device Attributes to include in the expression. You can add multiple Device Attributes to an expression.

In the example below, we are invoking the Flex83 math functions against a Device Attribute to create a different form of Custom Attribute. As you can see this is quite powerful.



It is important to understand that because your application depends on the time series data being sent from your devices, once the attributes are set this is the “Schema” of your database, and the database is safeguarded to maintain the integrity of your application. As long as you have “remaining” attributes available in your plan, you can continue to grow and enhance your Attributes.

But, there will be cases when you want to “reclaim” plan resources and remove elements you’re your schema, or simply change the way your application works. For example, you may want to send different data from you devices, create more derived data, any number of possible reasons to change your application and thus, your Attributes.

So, to reset your Attributes (in effect your database schema), navigate to the top of the Attributes page and click the arrow highlighted in red below:

Because this will in effect restart your entire data collection, you will be prompted to be sure you intend this action. But to change your schema you would proceed, and then start sending the new set of attributes from you devices (or simulations for that matter), and your new attributes will automatically appear here.

4.5 Connecting Devices to Your Flex83 Application

Because you may want to connect different kinds of physical devices, such as Arduino, ESP32, Raspberry Pi, etc., we will show several ways to connect to the Flex83 Application to make this as clear as we can in the following sections.

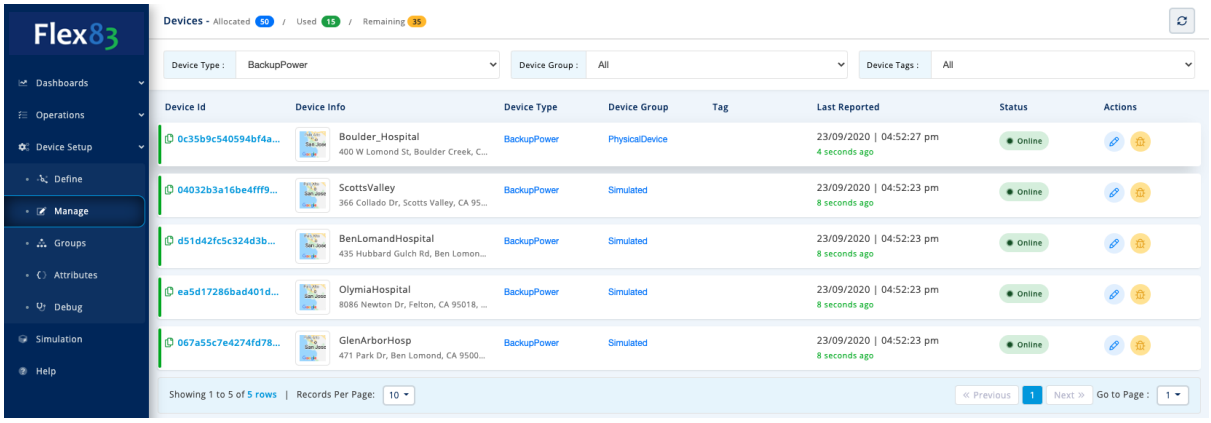
But as a tip for getting started with any new project, a powerful way to start is to set up your Device Type, with however many instances of that device type you want, and then create two (or more) Groups on that Device Type and put some of your devices in each group (I like to set up "Physical" and "Simulated" Groups to start). Then, you can go to the Simulator, and setup a simulation pointed at the "Simulated" Group, put the JSON you want into the simulator and give it a try. If you like it, great, move on to connecting physical devices. If you don't you can re-set your attributes (removing your schema) and use a different JSON. When you have what you want, you can add physical devices, and when you want, delete the simulation and deploy those devices as physical devices as well. This is a great feature for prototyping your initial design and getting started.

But before we start with any examples of adding physical devices, let's look at the Credentials File that Flex83 sent to our email account (when we clicked Email Credentials above). Note: if you don't immediately see the email be sure to check your spam folders before you click again!

Here are the contents of that file (I transposed the file because I think it is easier to read this way):

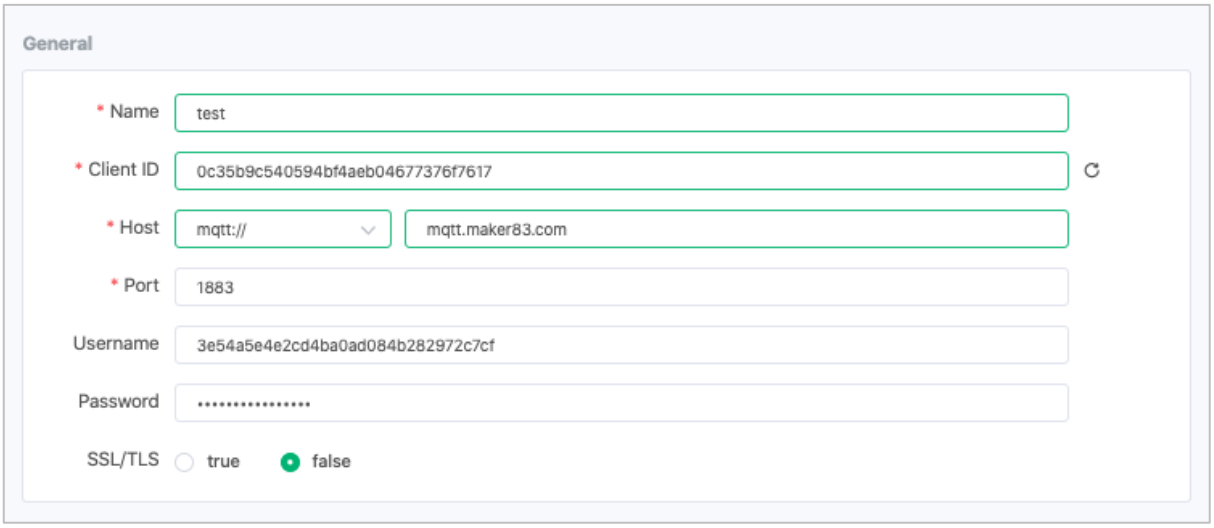
Scheme	**TCP**	**TLS**
Username	184c41e9a61345a6af8fb4e591a0b711	184c41e9a61345a6af8fb4e591a0b711
Password	amVRBa9w1VLpNNik	amVRBa9w1VLpNNik
MQTT Broker Url	mqtt.flex83.com	mqtt.flex83.com
Port	1883	8883
Publish Topic	d047ec12c81e4188ae977bfa4ecf4d7a/<deviceId>/report	d047ec12c81e4188ae977bfa4ecf4d7a/<deviceId>/report
Subscribe Topic	d047ec12c81e4188ae977bfa4ecf4d7a/<deviceId>/control	d047ec12c81e4188ae977bfa4ecf4d7a/<deviceId>/control
MQTT Root CA	N/A	https://dl1jn0gbol8q5.cloudfront.net/ca/RootCA-maker83.pem

This, plus the individual device ID for specific devices provides all the information we need to connect our devices. For clarity, the individual device IDs can be found on the Device Setup / Manage page, and you can copy the device IDs to the clipboard by clicking green icon just to the left of the ID string below:



4.5.1 Connecting using MQTTX

MQTTX is an MQTT setup and debug utility that makes it easier to connect to cloud applications for the first time. The idea is if you can get connected using MQTTX, you see how everything is supposed to be working, and you can map that into your embedded IoT solution. You can download MQTTX here (https://mqttx.app/) and install it. Once you launch it, you can click the "+" icon on the left to create a new connection. Create a name for the connection (any name will do) and then configure the connection as follows (note – replace mqtt.maker83.com below with mqtt.flex83.com or mqtt.iflex.com, depending IoT83 on the product you are using):

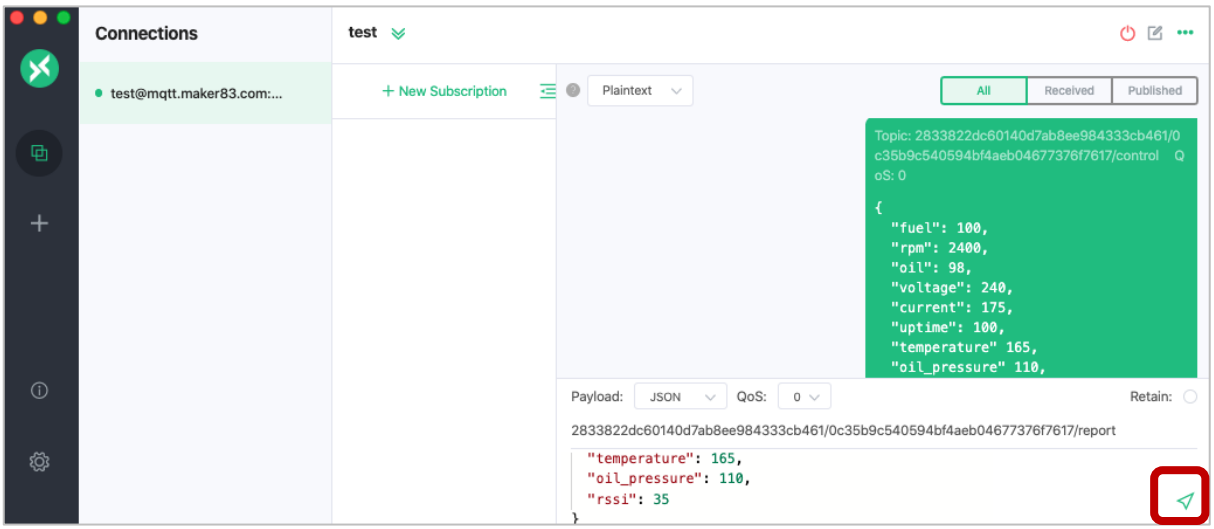


Here the Client ID is the unique client ID from above (Device Setup / Manage) for the device you are connecting, and for the host field, fill in from your credentials file the Flex83 MQTT url (mqtt.Flex83.com), add port 1883 (to connect without TLS, for this example), then add your user name and password from your credentials file. REMEMBER the username / password combination is different for each device type you have created, and you have a different credentials file for each! Finally select "false" for SSL/TLS.

In the next dialog box, "Advanced", you can just leave the defaults as is.

In the "Last Will and Testament" dialog box you add the MQTT topic as well as the payload for the MQTT publish transaction. The topic takes the form of <device type iD>/<device ID>/report, similar to what you see below. Finally, you can drop in a copy of the JSON for this device type, remembering that it needs to match the JSON that you are using to define the schema for this device type.

Once this is done, you can click “Connect” at the top of the screen, and if all goes well, you will see a screen like this (if not MQTTX provides error messages that help you identify your problems!).



By clicking the “arrow” on the lower right (circled in red) you can repeatedly send your payload to Flex83. At this time, if you navigate to the Device Setup / Manage page (and perhaps click refresh if you were already on that page), you will see that your device is now connected! Also note that the connected status will time out after a period, but you can always reconnect MQTTX and reissue the MQTT Publish actions to re-connect.

4.5.2 Connecting using Mosquitto

Mosquitto is a very popular MQTT application. There are multiple tutorials on how to install and run Mosquitto, so this explanation will just show how to connect and publish to Flex83 using the Mosquitto_pub function of the Mosquitto Client.

The general form of a Mosquito Publish Client request would be:

Mosquitto_pub -h <host ID> -p <port> -u <User ID> -P <password> -t <topic> -m <message> -I <Client ID>, where you replace "<XXX>" with the appropriate field similar to what we did with MQTTX. The only tricky thing here is that the JSON you send cannot have any "carriage returns", so you have to remove them so that the JSON is on one line. Though this is hard to read, the command line Mosquitto_pub command would look something like this:

mosquitto_pub -h mqtt.flex83.com -p 1883 -u 3e54a5e4e2cd4ba0ad084b282972xxxx -P wsCuhQuBHBT0xxxx -t 2833822dc60140d7ab8ee984333cxxxx/0c35b9c540594bf4aeb04677376fxxxx/report -m '{ "fuel": 100,"rpm": 2400,"oil": 98, "voltage": 240, "current": 175, "uptime": 100, "temperature": 165, "oil_pressure": 110, "rssi": 35}' -i 0c35b9c540594bf4aeb04677376fxxxx

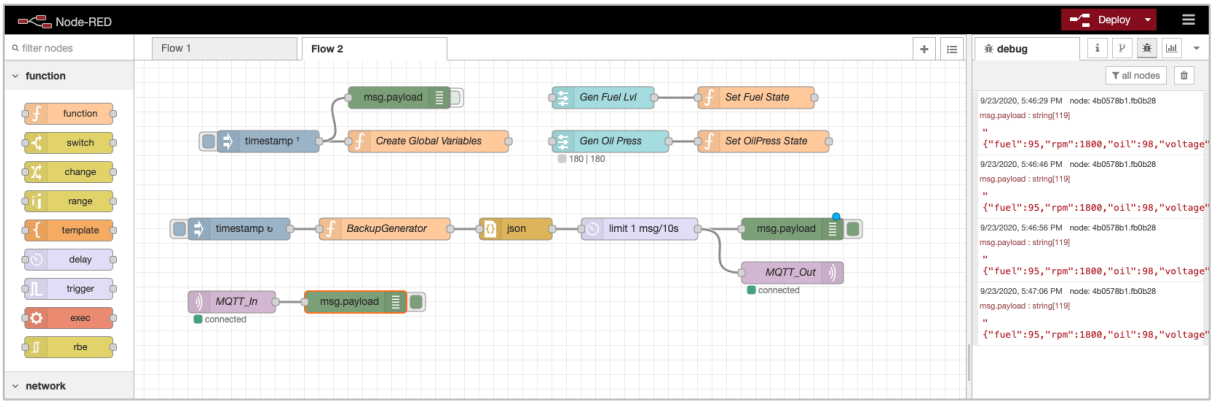
(Note: the "xxxx" above were to obscure the details of my account!!!)

When successful at the command line, you will receive no error message (if there are problems you will receive generally helpful error messages). And you can again navigate over to the Device Setup / Manage page and see (once you refresh) that your device is connected.

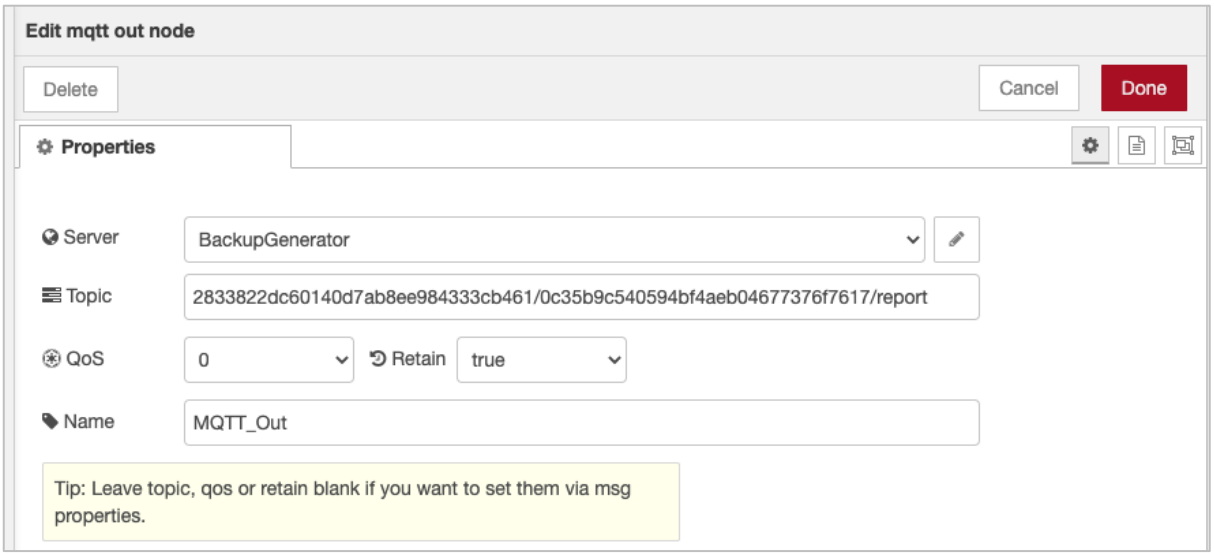
4.5.3 Connecting using Node Red

Node Red is a very popular tool for connecting sensors and devices on edge devices and also for creating Edge Computing in gateway devices. Here we will provide a few screenshots to show how to connect Node Red to your Flex83 application.

First, here is a snapshot of the Node Red flow we deployed:

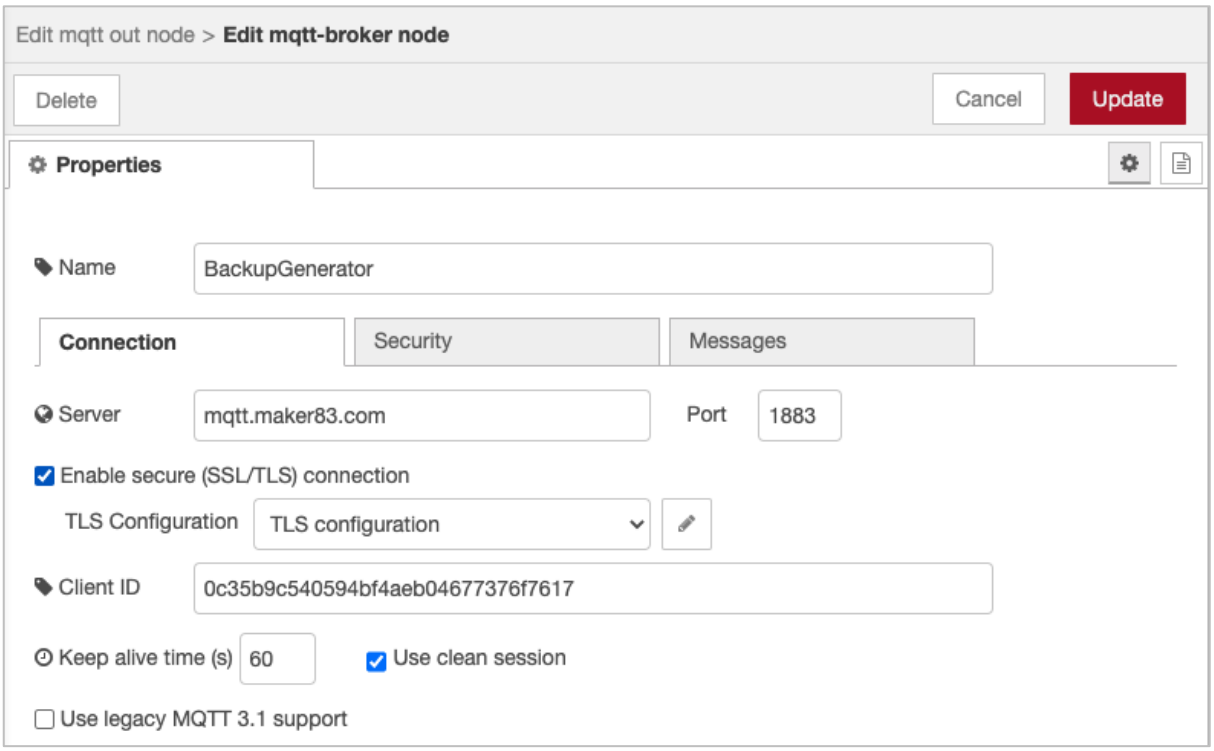


Here, a timestamp node (to trigger event generation) is connected to a function node that creates the data object we want to report, this gets converted to JSON by a JSON node, we limit how many events we send to the cloud with the limit node, and finally we send the data to the cloud via the MQTT node. While out of scope for this discussion, the other "Dashboard Nodes" are used to be able to dynamically change the value of several of the JSON attributes (with the values passed using node red's global value functions) for fault management demonstrations and to trigger alarms or events on Flex83 (using the Event Manager). Setting the MQTT node is done as follows, where this example also is implementing TLS. So, clicking the MQTT node you see this screen:

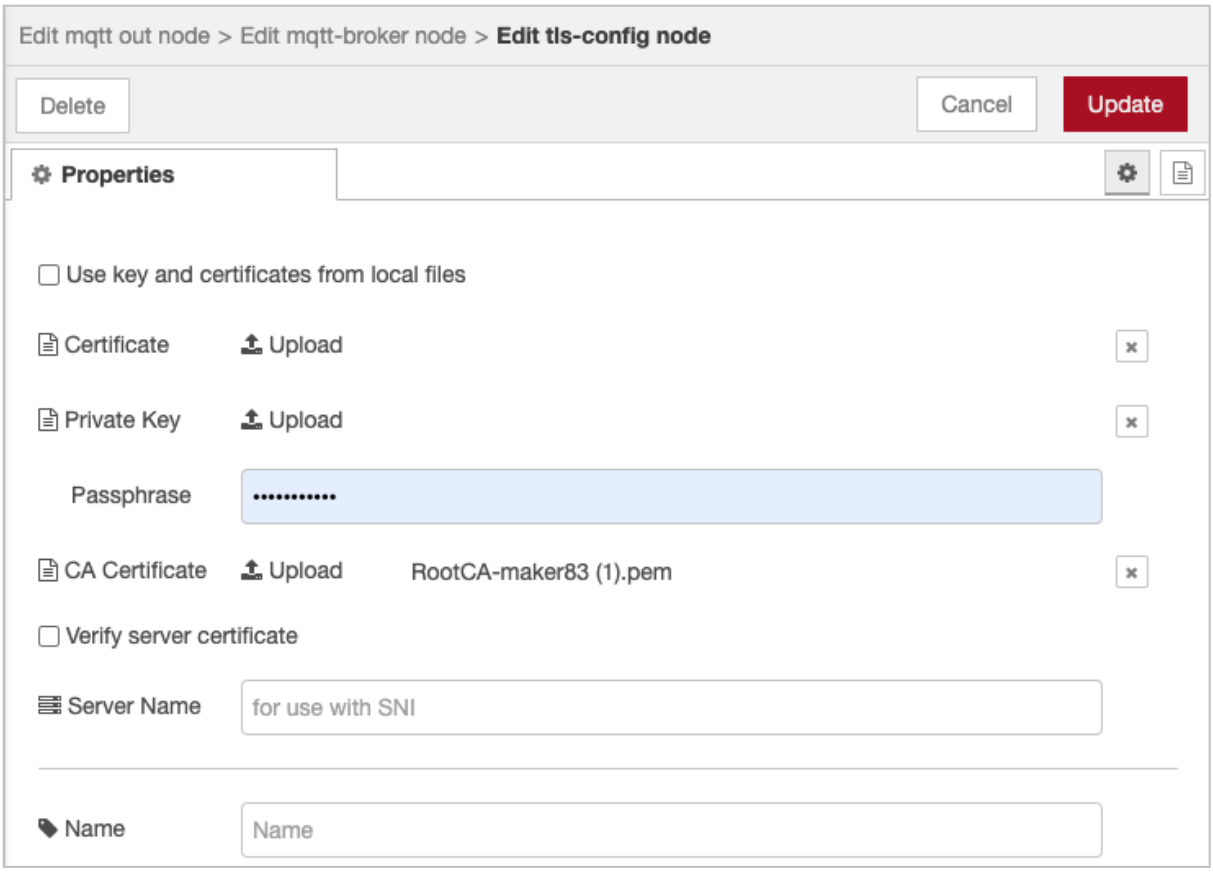


Here we have created a new MQTT Server Name, BackupGenerator (named to be similar to the Device Type), the topic is set up just like in the earlier examples (<Device Type ID>/<Device ID>/report), the QoS can be 0, with Retain = true, and the node is named to be easy to understand its purpose in the flow.

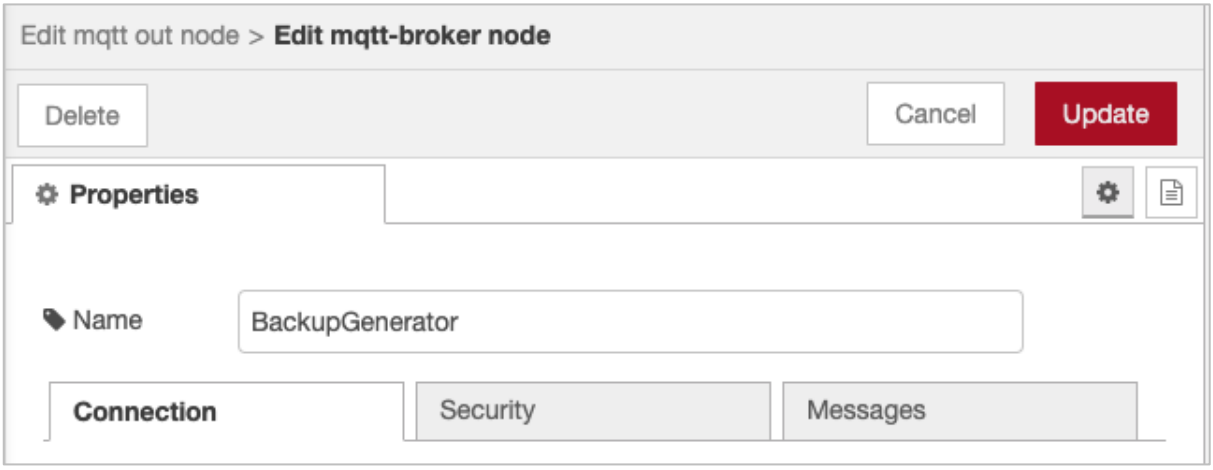
Then clicking the button to configure the server (the pencil icon next to “BackupGenerator”), we see:



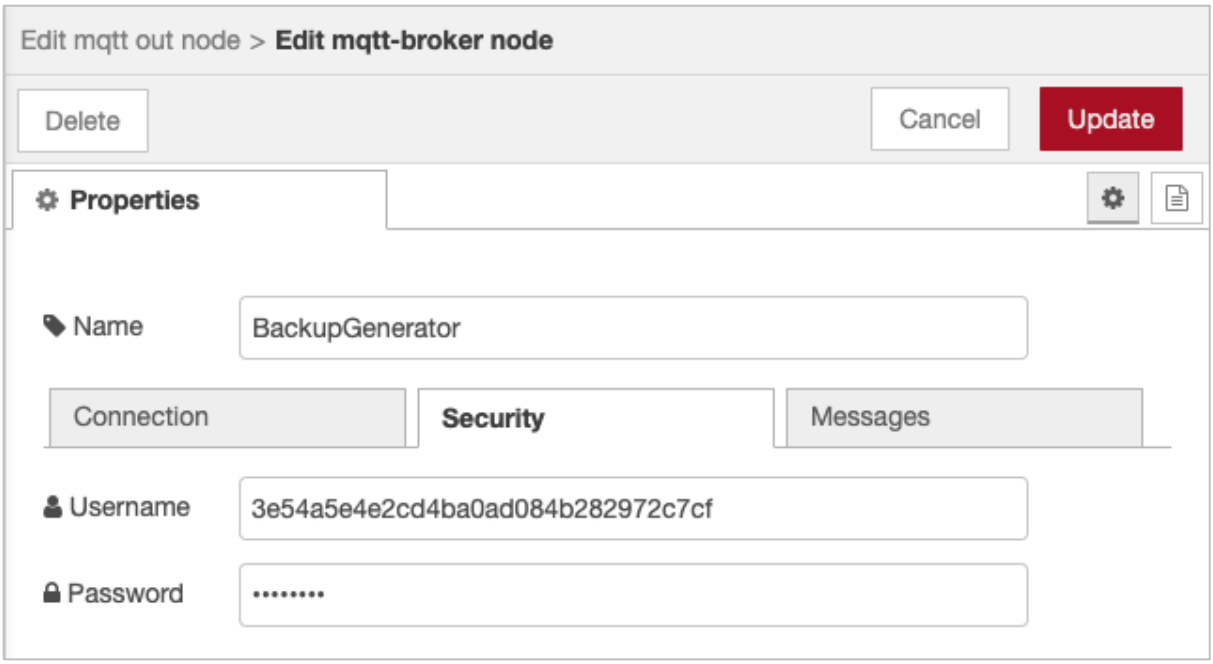
We add the Flex83 server as expected (where the above clip shows mqtt.maker83.com, use mqtt.flex83.com instead), with port 8883 assigned since we are using TLS, select “Enable secure connection”, Select TLS configuration, and add the Client ID for the unique device we are connecting in the Client ID field. Then we click the “pencil” icon to configure the TLS connection:



Here we just need to set up the CA Certificate (third down from the top, and note, this will be a flex83 certificate, not a maker83 one). To do this, we drop the MQTT Root CA url that we have from our Credentials file into a browser, hit enter, and the CA file is downloaded into our downloads folder. We click upload next to CA Certificate in this form, and we are good to go here. Click Update to exit this form, which takes us back to the prior form, where we select the Security Tab:

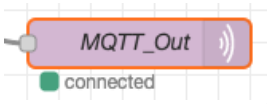


And now see this:



Where we add our MQTT username and password from our credentials file (remembering that each Device Type as a different credentials file, and using the right one!).

Now we click update here, and update on the prior form, and then Done on the final form, and the Deploy our updates to Node Red.

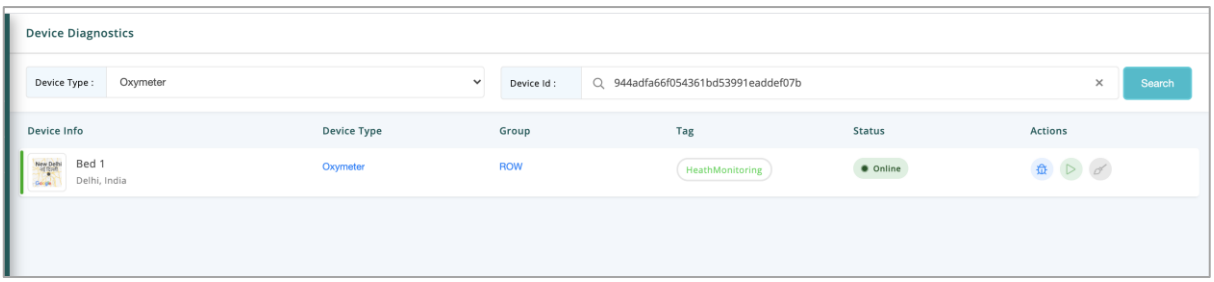
Assuming all is well, next to the MQTT node we see: 

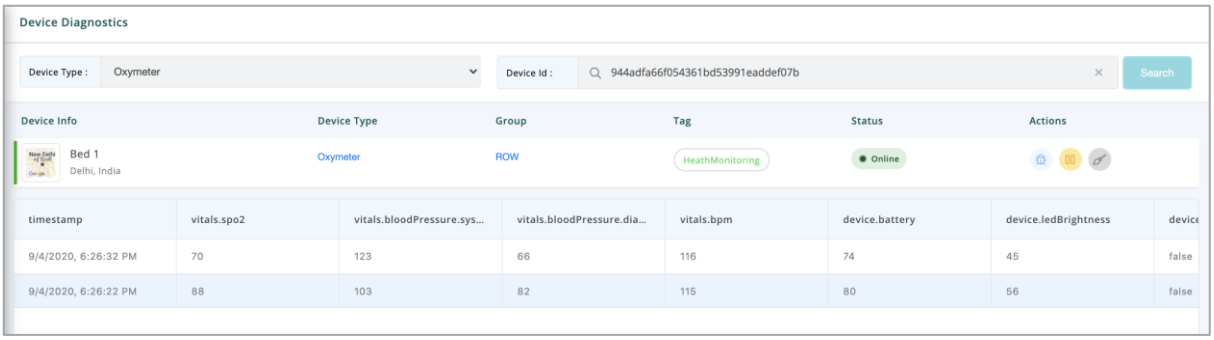
, and we can navigate to Device Setup / Manage and see that this device is connected.

Now that you are connected using Node Red, there are all sorts of things that you can do, from connecting new sensors to pins on Raspberry Pi's to building edge control functions and more. Also note that the MQTT "in" configuration works the same way and with that we can receive Remote Procedure Commands from Flex83 via the Event Manager application or via manual injection on your dashboards.

4.6 Debug

The Flex latform gives you the ability to debug or sniff RAW MQTT packet coming from your Device. Do this from the Device Setup / Define menu, or from the Device Setup / Debug menu:





By this point, you are done with the basic set up of your Device.

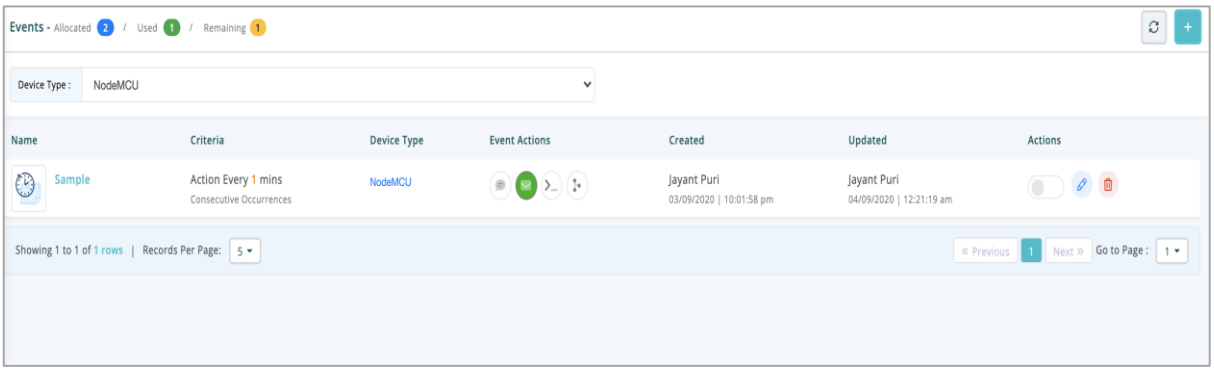
We could now jump directly to our Dashboards to see how our application has been dynamically growing behind the scenes throughout our configuration actions – but first, let's take a look at the Operations Menu items and configure those as well.

5 Operations

5.1 Events

The Events capability on Flex83 is very powerful. Here you can configure actions based on either simple threshold crossings or more complex combinatorial events monitored on the telemetry data from your Devices, or even from you Derived or Synthetic Attributes that you have created. Event can be anything for example, If, the temperature in a "cold storage warehouse" goes beyond 10 Degrees, for (say) 5 consecutive times, take an action – Actions can be Sending an email, sending an SMS, invoking a Remote Procedure Call command to the device or launch a webhook.

Take a look below to see how to configure Events for your Device. Click Operations / Events to see a screen similar to the following (initially you will not have any events, so click to "Create New Event"):



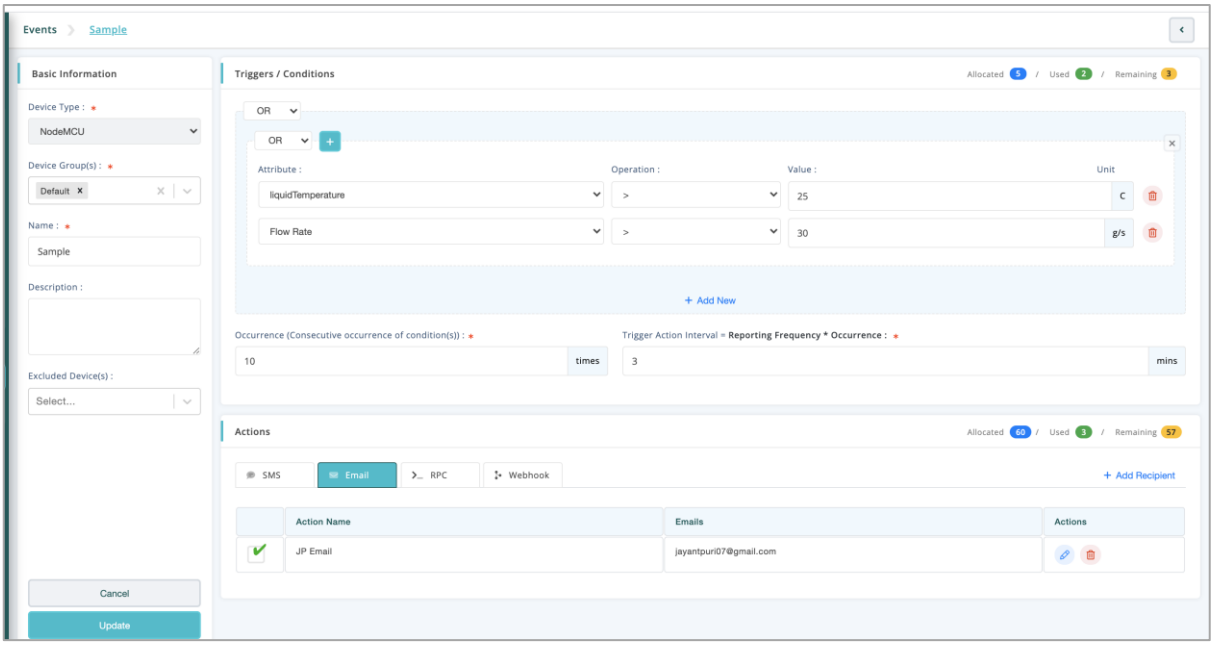
At the top right, click the "+" to add a new event, and you will see a screen like the following.

On this screen, we first set the Device Group(s) that this Event applies to, name the event, create a description, and identify any devices that should be excluded from this Event's evaluation policy.

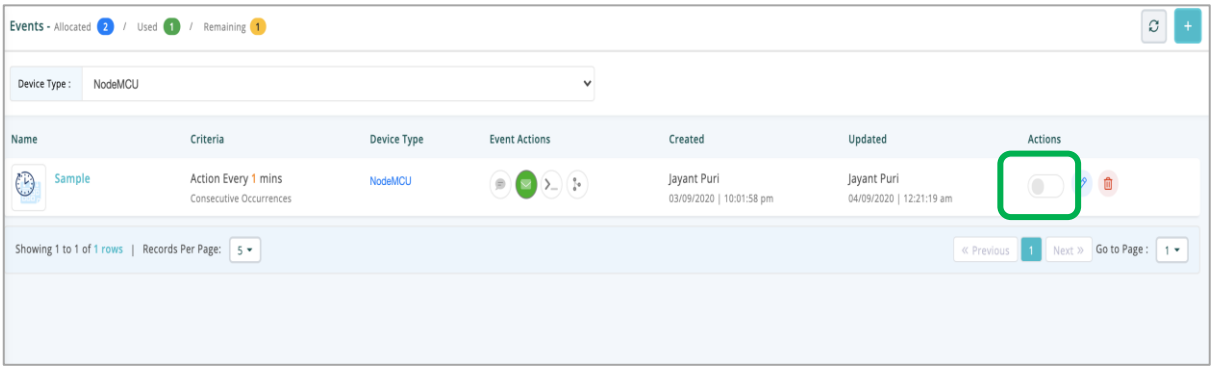
Then to set up the Event Monitor, we are configuring to trigger the Event when "liquidTemperature" is greater than 25, OR "Flow Rate" is greater than 30. When this is true for 10 consecutive times the event will trigger. (Using 10 consecutive times is just a way of saying that we only want to trigger if this condition is true for 10 times the data update frequency). Finally, to avoid activating this Event constantly, we set the trigger action interval to 3 minutes (this is like a time out period to give us a

chance to react to the event and to not be overwhelmed with Event messages). Finally, the “Action” for this event is to send an email.

We have clicked the edit icon on the lower right of this screen to edit the email messaging for this Action. Finally we check the box to use this Action, and click Update to save the event.

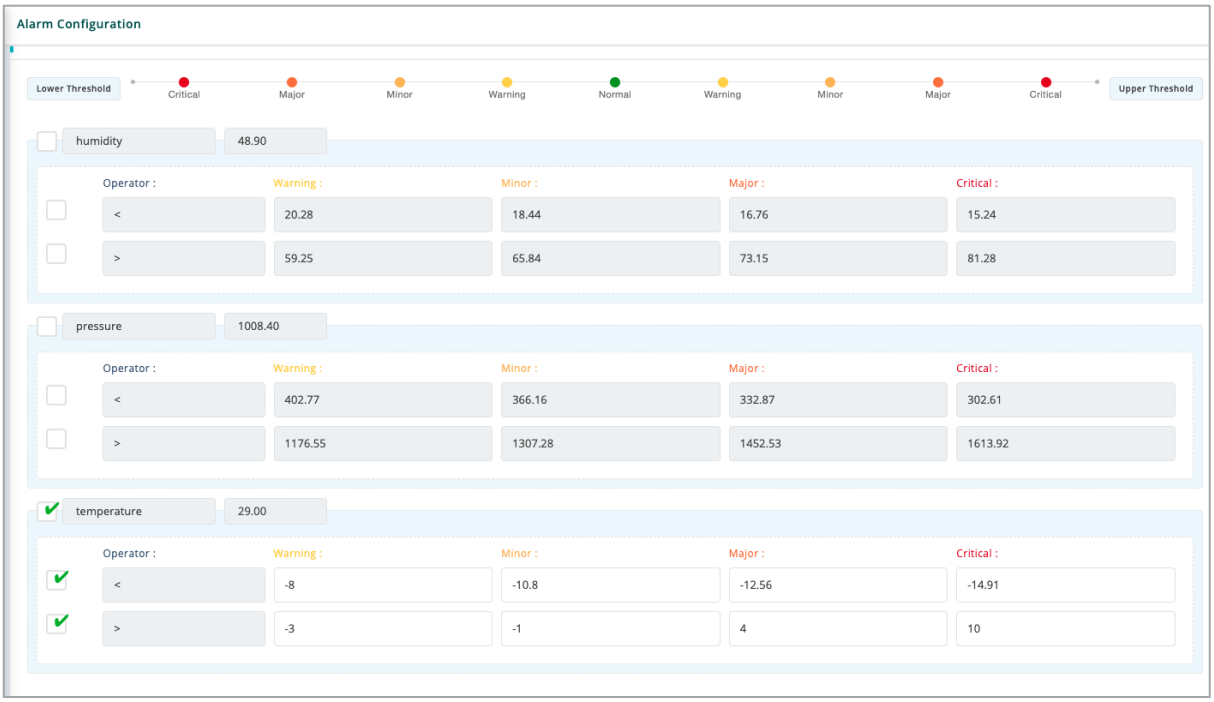


Now when we return to Operations / Events, we can turn on this Event by sliding the toggle highlighted in green below:

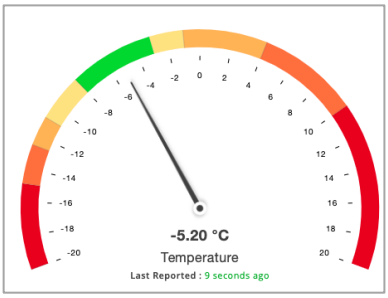


5.2 Alarms

This section allows you to set value based thresholds for increasing or decreasing trend of your values. Usually, you want to track if a telemetry value is trending too high, or too low, for example if a pressure reading is operating above threshold, or if a power measurement is trending too low. But the Method83 Alarms management solution also allows you to track telemetry data to a “sweet spot” and simultaneously track deviations both above and below ideal value ranges. For instance, if your device is monitoring temperature for a Cold Chain, then probably you would want the temperature to stay between -8 °C and -3 °C beyond which (like too cold or too hot would cause the items to be damaged so you would want to control that by configuring correct thresholds. The configuration below (with the green check marks) accomplishes this:



Further, once these setpoints have been established, when you create monitoring gauges on your device drill-down pages, these setpoints, Warnings, and Alarms are all indicated on your Dashboard Gauges. For example, the alarm settings for temperature above, would result in a customized gauge on your drill-down monitoring Dashboard as follows:

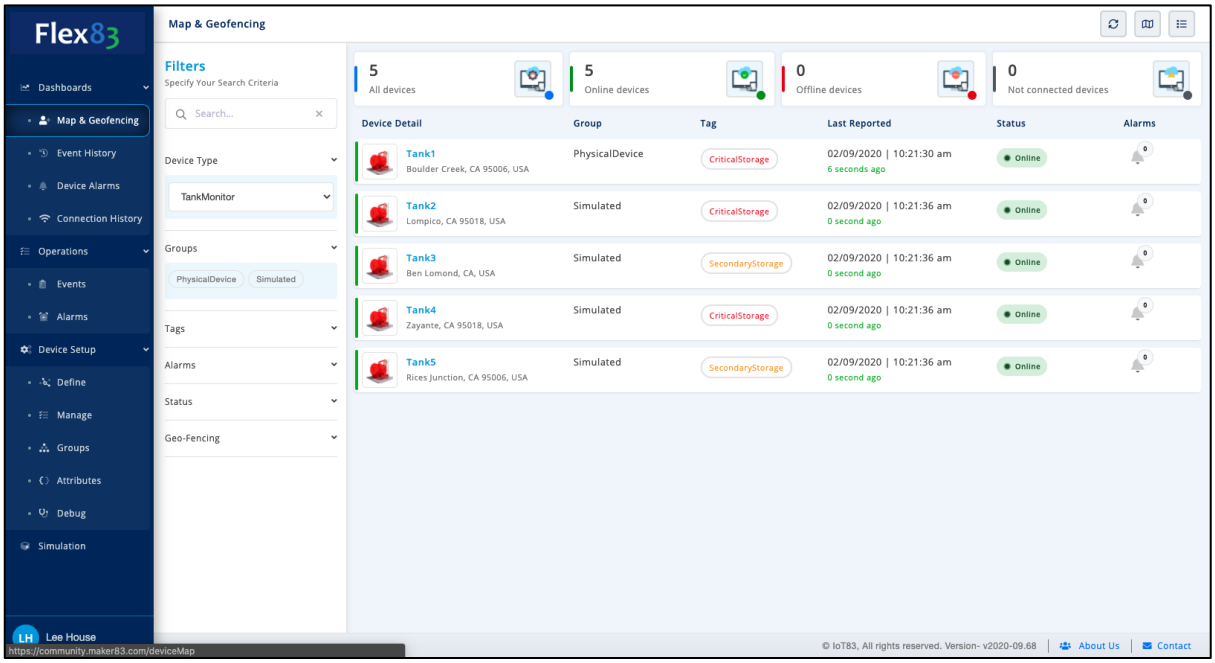


Now that we have completed the Device Setup and Operations sections, we can turn out attention to the Dashboards and the Application itself and see what the setup has produced for us.

6 Dashboards

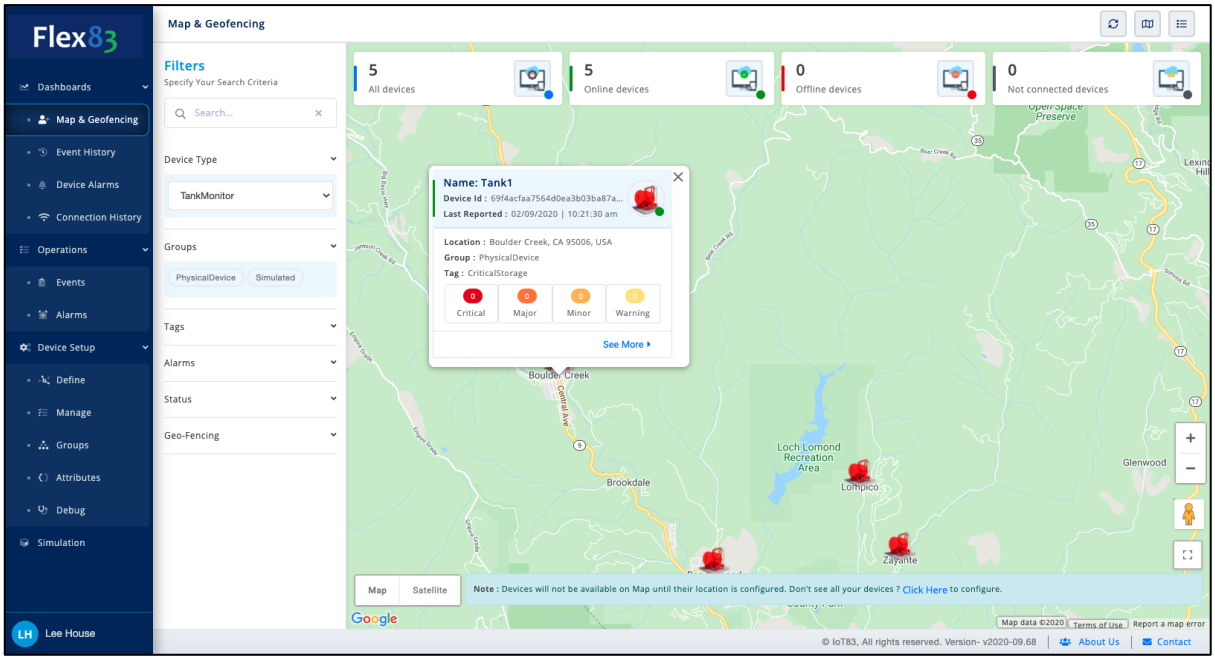
Now, we shall look at the Dashboards that are automatically created once we configure our Device Type & Operations.

6.1 Device List



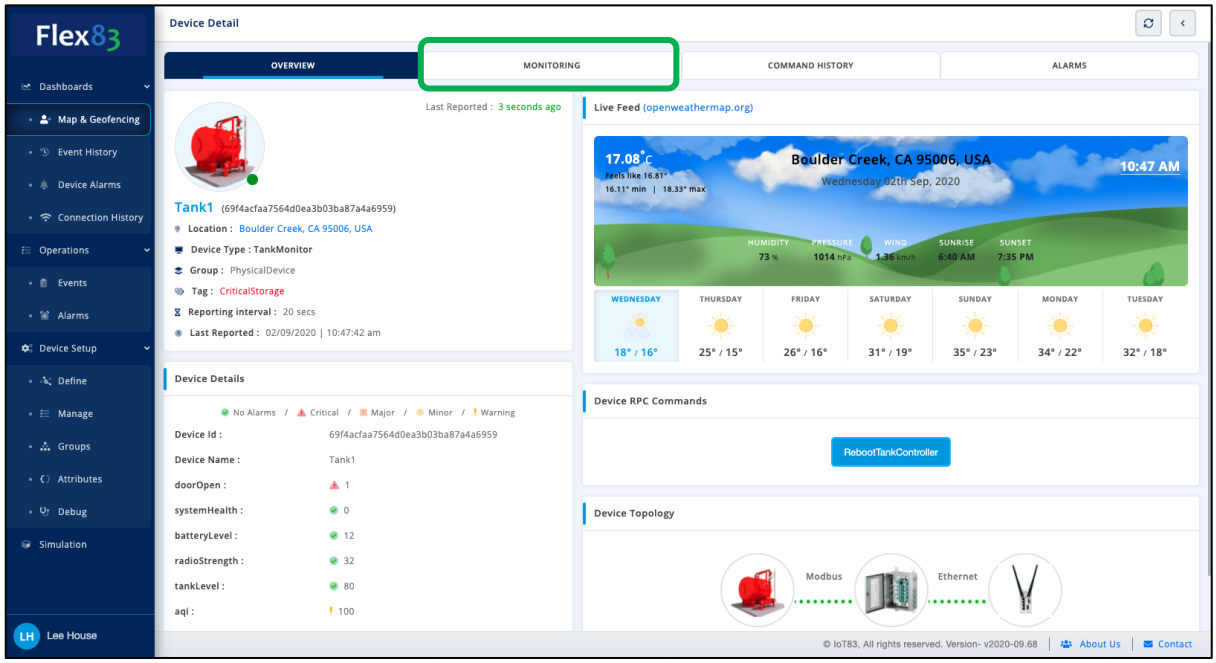
The Device List provides a summary view of each of your devices and provides a path to “drill-down” to detailed information on each device.

6.2 Map & Geofencing



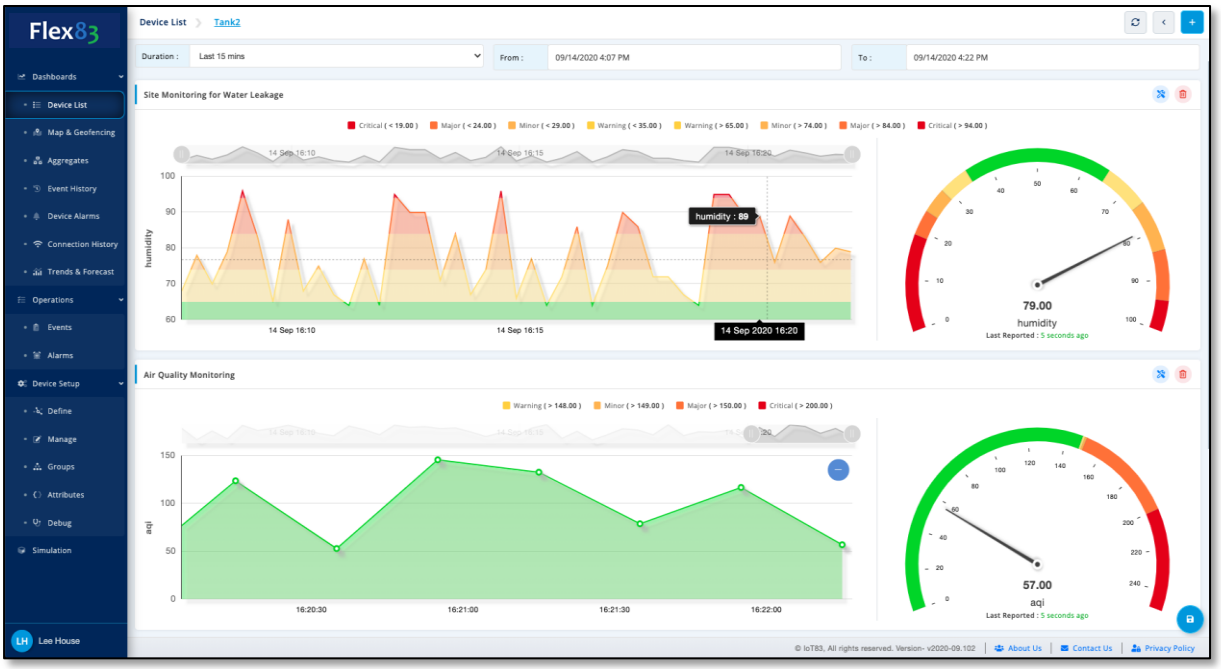
Similar to the Device List view, the Map & Geofencing view provides a summary view of each of your devices and provides a path to “drill-down” to detailed information on each device. But by using the Geofencing feature you can also identify when any of your assets are withing a certain settable proximity of a location you identify.

And, by clicking the “See More“ section on the Map or the Device name on the list page, page you can drill down to the device details page. Below are the screen shots for the same.

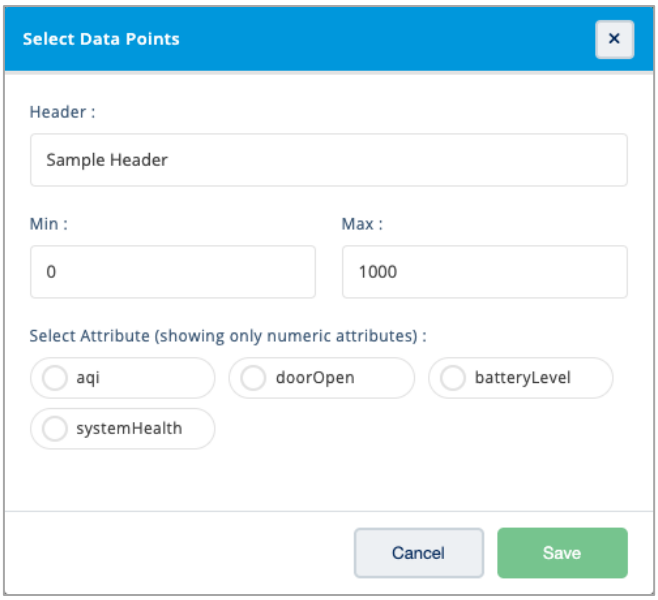


This drill-down view provides a great deal of information about a particular device: the configuration information for that device, its tags, the telemetry data it is sending, its, network layout, as well as the ambient conditions at the site of the device.

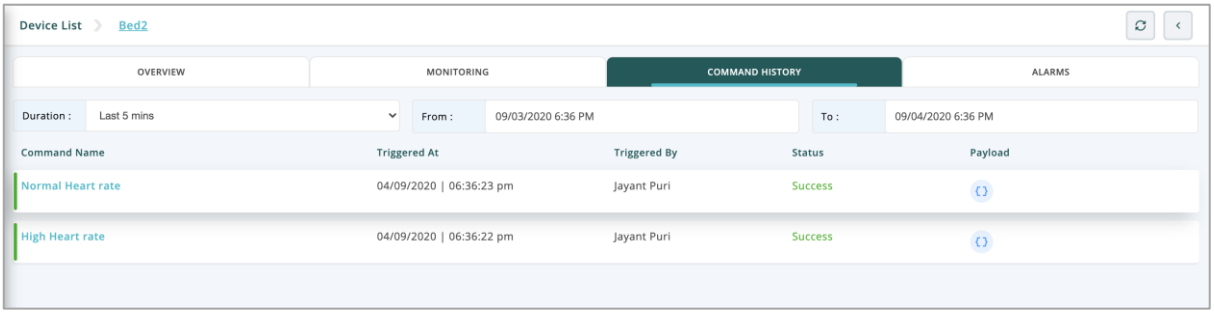
We can drill down further by clicking Monitoring, highlighted in green above, to build and then use screens like the following:



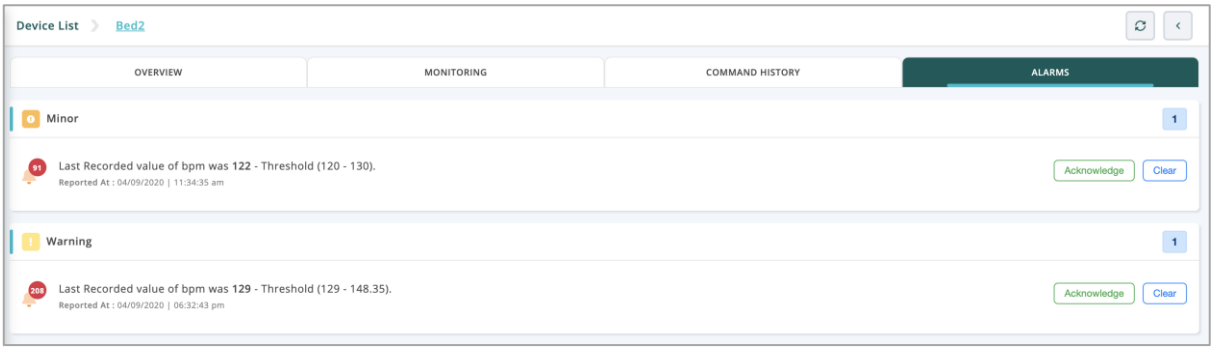
When we go to this screen for the first time, it is blank, because we have not yet set it up – but that is easy. To get started here, just click “+” icon at the top, and you get a configuration screen that looks like this:



Here you can name this section of the Monitor Dashboard, set minimums and maximums for the chart elements and then select the Device Attribute you want to add to the page. You can continue with this process until you have added all of the Device Attributes you would like to monitor.



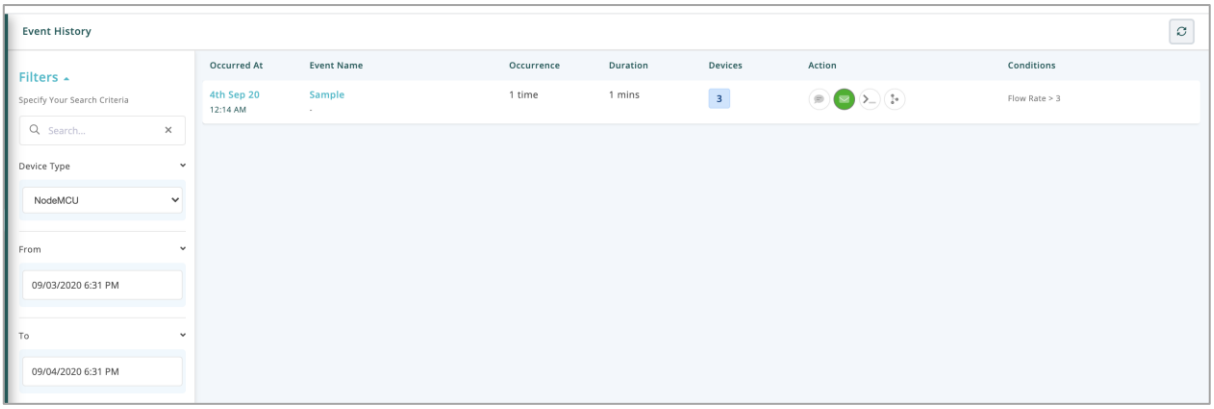
The Command History tab provide you with a record of all RPCs that have been executed by the system, either via the system management console, or via Event triggered activity.



Finally, the Alarms tab provides you with a summary of all of the alarms that have been detected. (We will cover the Alarm Management tools of Flex83 shortly).

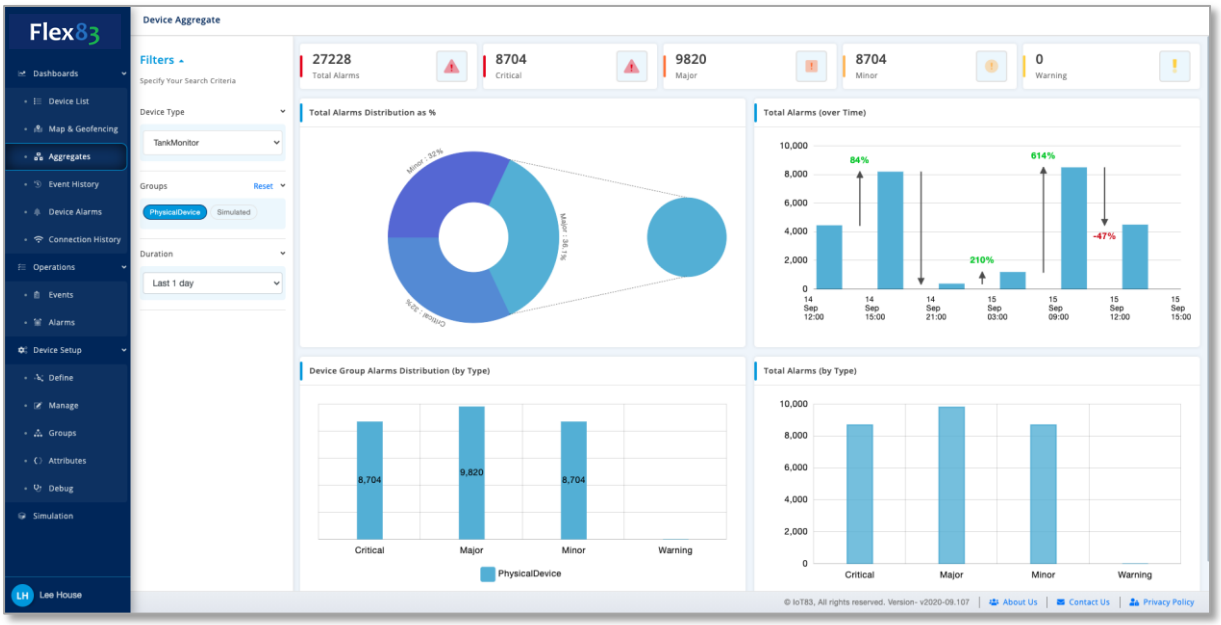
6.3 Event History

We will talk about events in the Operation section – this page give the historical view of all the Events that occurred for this device Type.



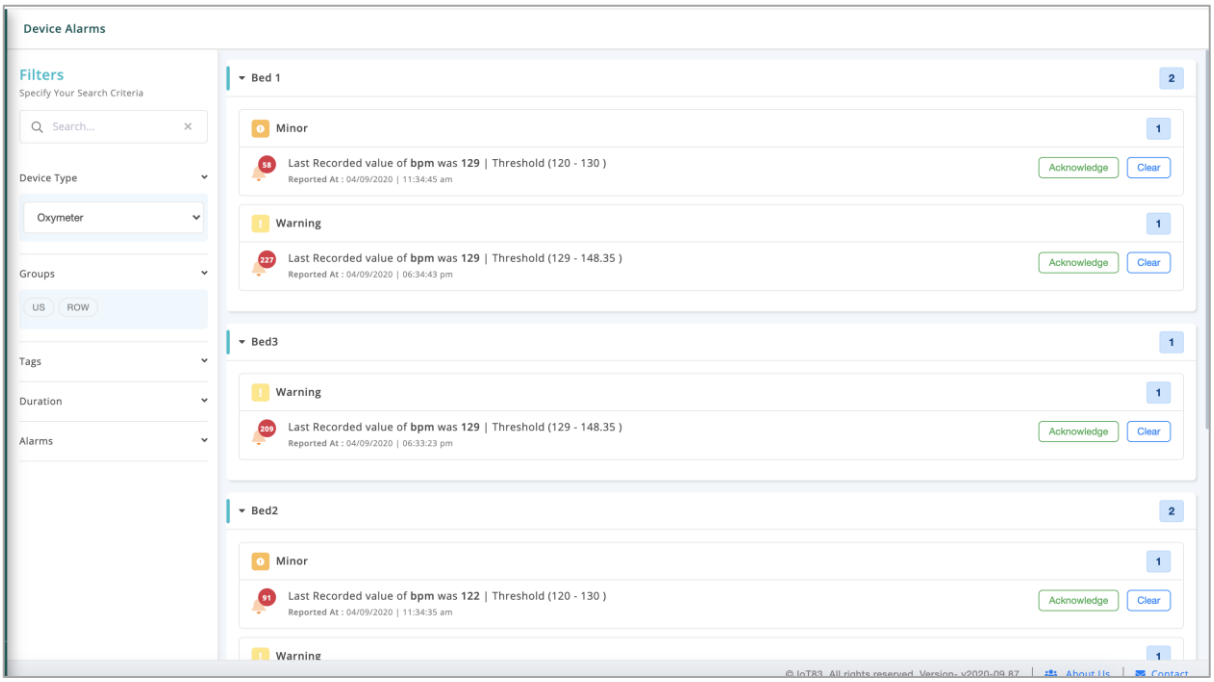
6.4 Aggregates & Device Behaviour Analytics

The “Aggregates” dashboard provided rich analytic of the behaviour of your solution from an alarms and alerts perspective. First, on the left you can filter for the device type, group, and time window you want to view. Then, across the top of the dashboard you have a summary of the total alarms by type that have occurred. Below, you can explore the percentage of alarms by type, the trending of alarm over time, total alarms by type, and the distribution of alarms by type. This analysis can be applied across your device types and groups, across different windows of time to provide excellent insight into the performance and behaviour of your application.



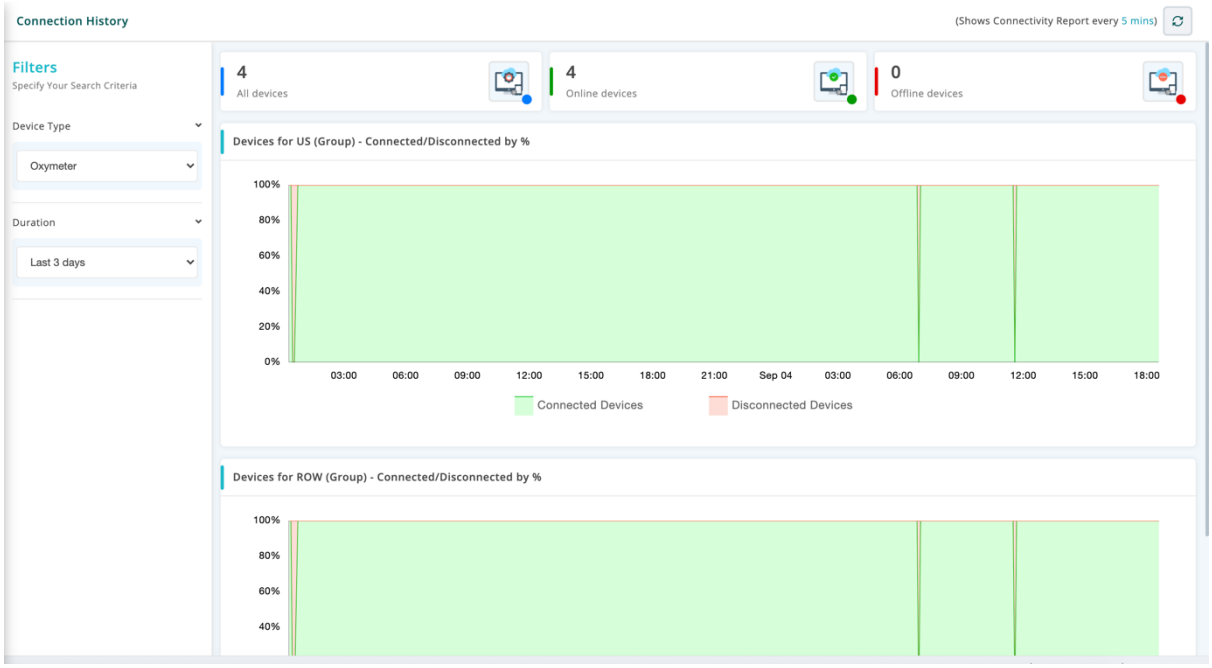
6.5 Device Alarms

This page shows all the devices that have alarms (of any type). You have a rich set filters to selected what needs to be viewed.



6.6 Connection History

This section shows you the historical Device Connectivity for all your devices. It captures the time intervals when the device was reporting versus when offline.



6.7 Trends & Forecast

The Trends & Forecast dashboard functionality is coming soon!

7 Identity Access Management (IAM)

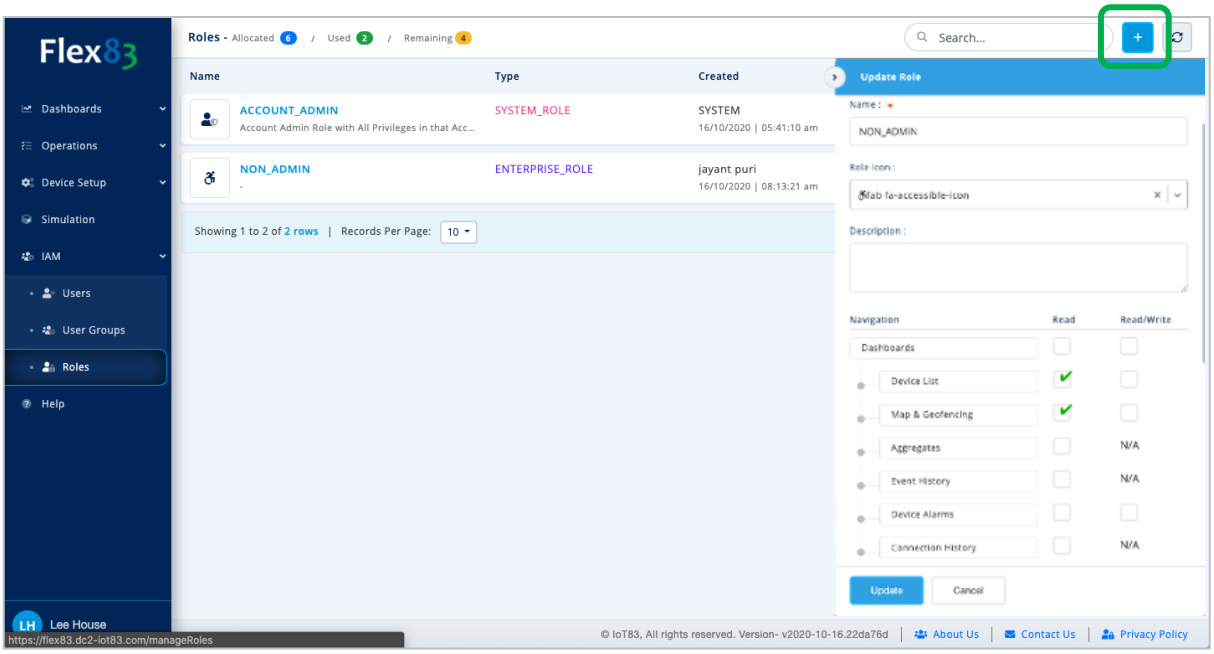
Flex83 is equipped with a powerful yet easy to configure and manage Identity Access Management solution. Using the IAM features, you can easily add end-user accounts for your application, assign these users to roles to better manage the permissions, data, and applications different user types

should have access to, on a role by role basis, and finally, the ability to create user groups to allow cross sections of your users to be assigned to different projects, assets, or other logical user access and responsibility assignment management.

Because we want to add new users to groups and assign them roles, we will first show how to add roles, then groups, then new users where we will use the roles and groups in user on-boarding.

7.1 Creation of Roles

The screenshot below provides a view of the Role Management screen. Like other Flex83 workflows, the "+" button at the top right allows you to add a new item, this time, and new user:

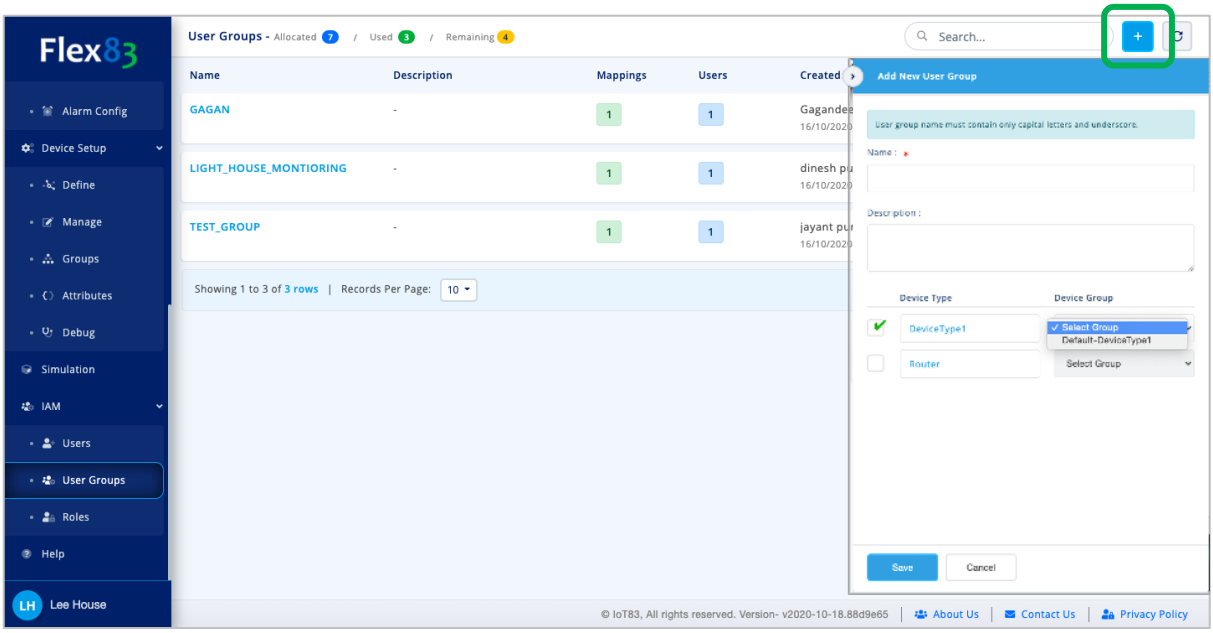


When you add a new role, you see a dialogue box as shown above. Simply fill in the new role name, add an icon to the role to aid in managing role context, describe the role scope, and then select the dashboard access capabilities you want to assign to the role. This last step allows you to re-use pieces of your application across multiple roles, but also have some roles that are uniquely qualified, or authorized, to access certain components of your application. Using this feature users see what they need to see, but not what they don't need, or should not have access to.

Typical roles may include admin (with full access to your application), Managers, Technicians, Engineers, Service Agents, etc.

7.2 Creating New Groups

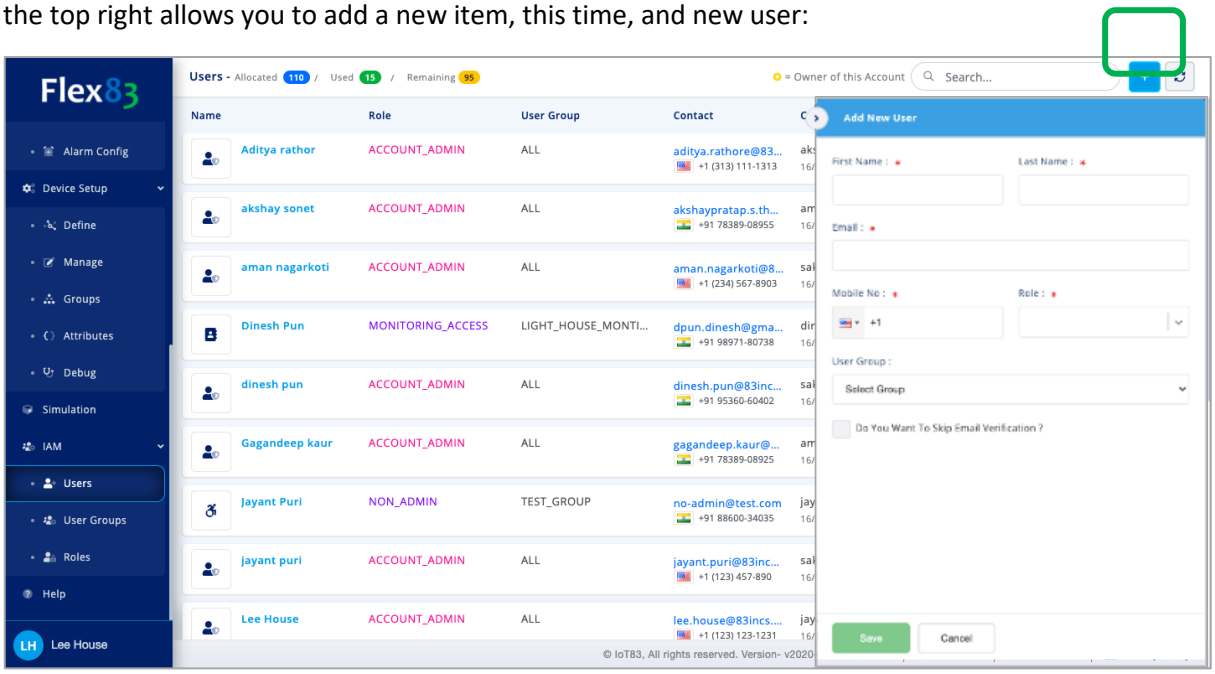
The screenshot below provides a view of the Group Management screen. Like other Flex83 workflows, the "+" button at the top right allows you to add a new item, this time, and new user:



Similar to creating new Roles, just fill in the dialogue box with the name of the Group, a description of the Group (for example, Users supporting Upper Manhattan HVAC systems, Users supporting Tenant X, Users involved in project Alpha, etc.), and finally select the Device Types that this Group will have access to.

7.3 Creation of Users

Now that we have created Roles and Groups, we can move on to creation of users. The screenshot below provides a view of user management page. Like other Flex83 dashboards, the "+" button at the top right allows you to add a new item, this time, and new user:



When you add a new user, you see a dialogue box as shown above. Simply fill in the dialogue box fields on the new user information and Save.

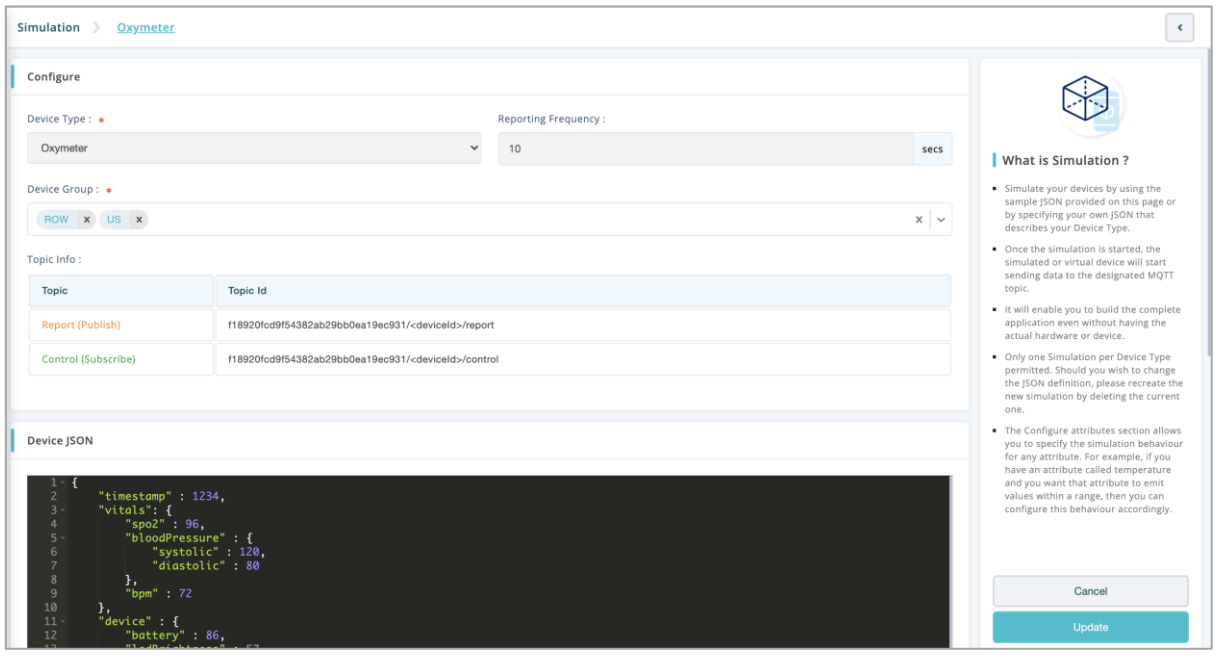
8 Simulation

This section enables you to create a “virtual device” and test both incoming and outgoing traffic to that virtual device. This can be very valuable for prototyping IoT application concepts prior to deploying hardware, to build up expertise with what is possible using the Flex83 platform, as well as to try different experiments with the Flex83 platform.

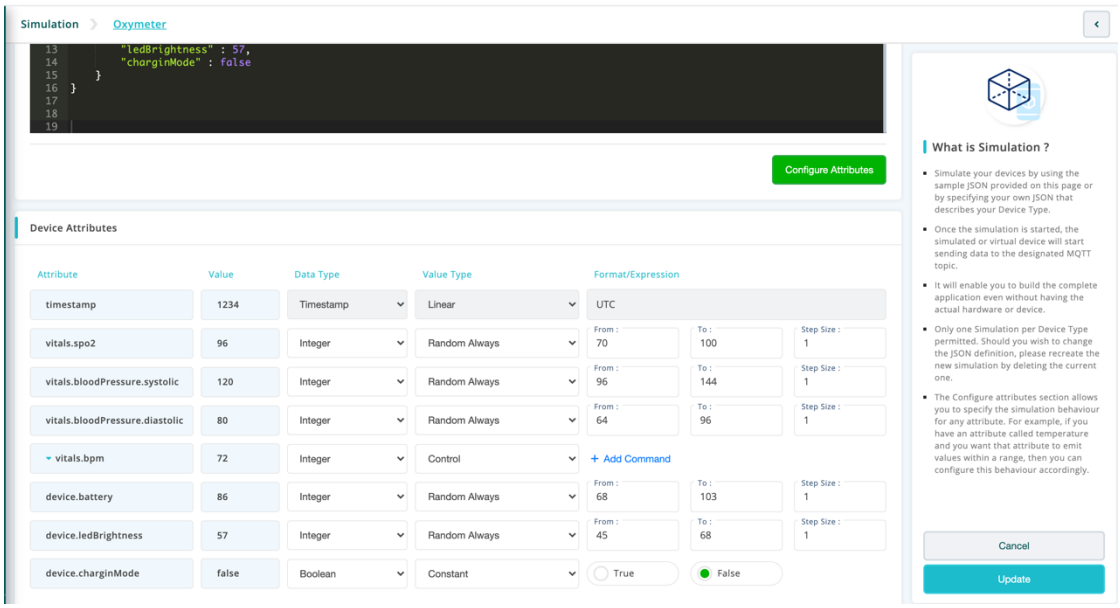
8.1 Creating a Simulation

Note, you can only create a simulation if you have already completed the Device Type creation steps.

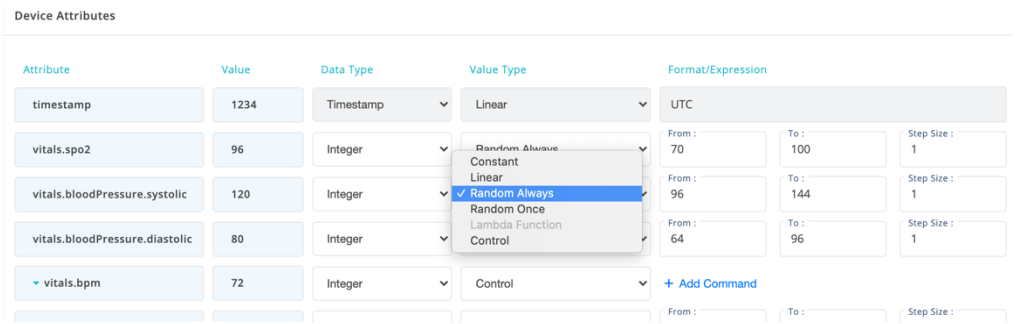
To get started, click “Simulation” and click “Add New Simulation” (or click edit on the an existing simulation to edit that). As a first step, select the Device Type that you want to Simulate.



The select the reporting frequency that you want the simulation to run at. Then add Device Groups that should be fed by the simulation. Then add the Device JSON that you want to use in your simulation. Note that there are several example JSONs provided if you want to try those to start. Once you have added the JSON, click “Configure Attributes”, and then scroll down to the Configure Attributes section (below).

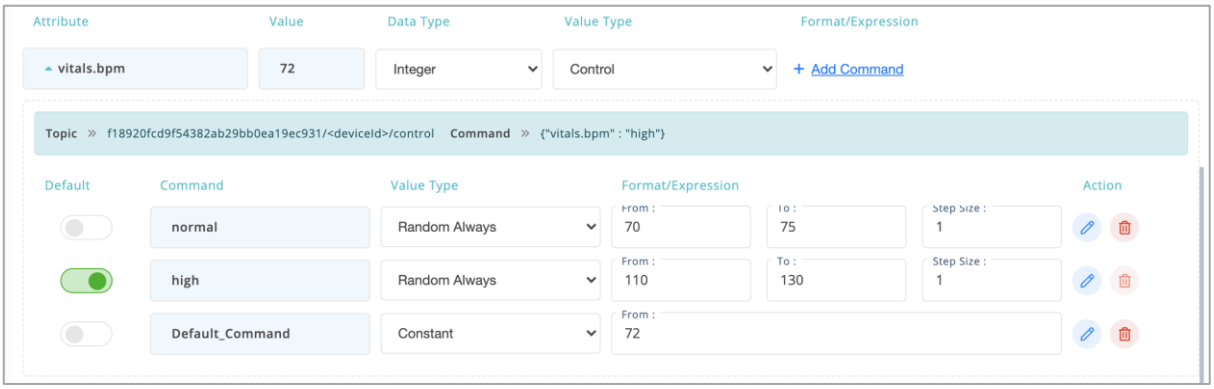


In this section, you can configure how the simulation will treat the different data elements or attributes. For each data attribute, we can configure how it will operate in the simulation. Let's take a look at our options:



For example, we can configure an attribute to simulate as "random always" in each cycle of the simulation, and set the possible ranges for the randomization, and the steps that it can increment. Looking at each option:

a) Constant will let you only pick a constant value for that data type.
b) Linear, will continuously increase that value as a liner function of time.
c) Random Always and Random Once, will behave randomly within a range of selected values either at the start only or throughout the simulation respectively.
d) Control is a special simulation behaviour which enables you to mock the Device behaviour on an Remote Procedure Call (RPC) trigger. For example let us assume that you have a device which can be rest or rebooted using a Remote Command (that is a JSON), so that behaviour can be configured in the system and you can define your commands. Sample commands for the Simulated Device are show below, let us examine this option in a bit more detail:



9 Iterations and Improvements of your Applications

The Flex83 solution has been designed to make it easy to build your first application, but also to continue to add to, update, modify and enhance your applications over time. So continue new devices, refining your dashboards, adding new logic and alarm and alert rules – all using the same tools as outlined above – all to make your IoT applications as rich and powerful as possible!

Also note, should you feel that you need the advanced features of the Flex83 Enterprise edition, please reach out to us at https://iot83.com/contact-us.

10 Help and Support

We at IoT83, and the engineers and designers of the Flex83 solution welcome your feedback on how we can make our solution even more powerful and useful to the community.

We are reachable at support@flex83.com