

# SPRINT 4

## Project Execution and Testing

Date	14 NOVEMBER
Team ID	PNT2022TMID35105
Project Name	Smart waste management system for metropolitan cities
Points	20

Run the Given Python code and check the

Web UI : [\*Smart Waste Management\*](#)

Location of Bins : [\*Location\*](#)

Wokwi code : [\*sensor - wokwi\*](#)

### Python Code for Bin 1

#Bin 1

```
import wiotp.sdk.device
import time
import random
```

```
myConfig = {
    "identity": {
        "orgId": "fzv53v",
        "typeId": "Bin",
        "deviceId": "Bin_1"
    },
    "auth": {
        "token": "1234567890"
    }
}
```

```
def myCommandCallback(cmd):
    print ("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']
```

```
client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()
```

## EXECUTION

- ✓ Executing the code to find the location of the Trash bin with random latitude and longitude which sends the data to the IBM Watson IOT platform

```
bin 1 - py (3.7.9)
File Edit Format Run Options Window Help
import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "fzv53v",
        "typeId": "Bin",
        "deviceId": "Bin_1"
    },
    "auth": {
        "token": "1234567890"
    }
}

def myCommandCallback (cmd):
    print ("Message received from IBM IoT Platform: %s" % cmd.data['command'])

client = wiotp.sdk.device.DeviceClient(config=myConfig, lc
client.connect()

def pub (data):
    client.publishEvent(eventId="status", msgFormat="j
    print ("Published data Successfully: %s" % myData)

while True:
    myData={'name': 'Bin_1', 'lat': 13.092677, 'lon':
    pub (myData)
    time.sleep (3)
```

- ✓ The data from the Python IDLE is received by the device created in the IBM Watson IOT platform where the latitude and longitude of bin is shown below

The screenshot shows the IBM Watson IoT Platform dashboard. The main table lists devices, with 'Bin\_1' highlighted. Below the table, the 'Recent Events' tab is selected, showing a stream of data events. The events are JSON objects containing device name, latitude, and longitude.

Event	Value	Format	Last Received
status	{ "name": "Bin_1", "lat": 13.092677, "lon": 80.188314 }	json	a few seconds ago
status	{ "name": "Bin_1", "lat": 13.092677, "lon": 80.188314 }	json	a few seconds ago

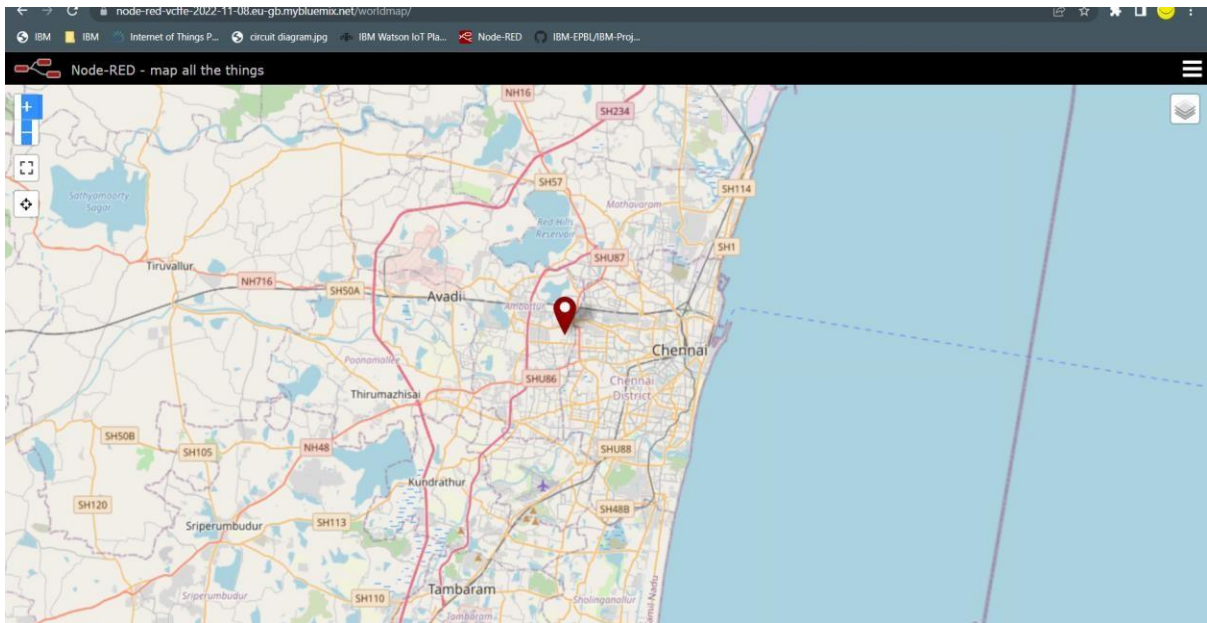
1 Simulation running

- ✓ The connections are made between the nodes in Node-RED Services to create web UI of the location

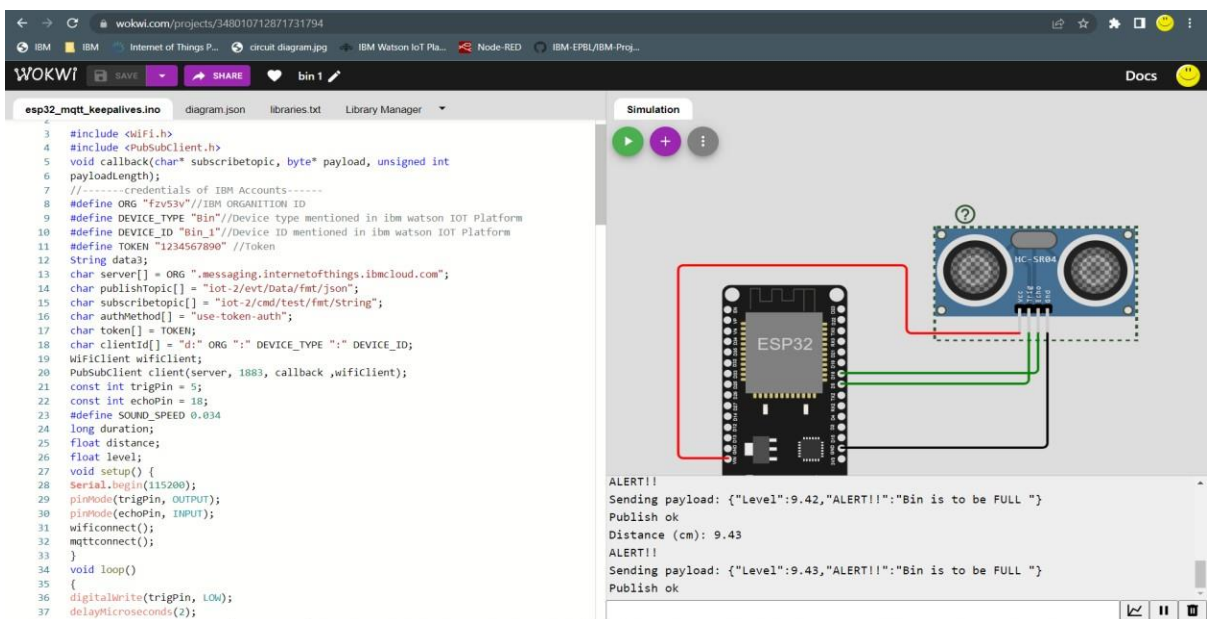
The screenshot shows the Node-RED interface. A flow is created with the following nodes: 'IBM IoT Bin\_1' (connected), 'msg.payload' (message node), and 'worldmap' (connected 1). The 'worldmap' node is configured to display the location data received from the IoT device. The debug console on the right shows the JSON payload being received.

```
{ "name": "Bin_1", "lat": 13.092677, "lon": 80.188314 }
```

- ✓ The location of the bin is shown in The World Map

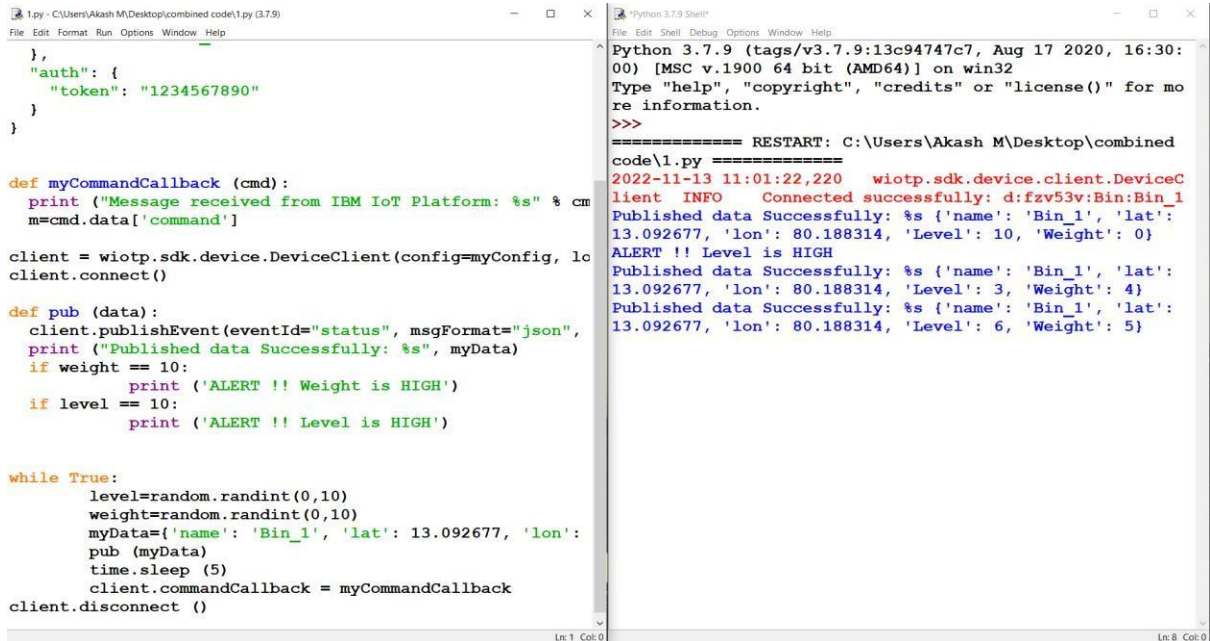


- ✓ The SENSOR values sent to the IBM Watson IOT platform by two ways:
  - From ESP32-Wokwi
  - From Python IDLE-Random Values
- ✓ To send sensor readings from ESP32-Wokwi



Run The ESP32 code Here : [Wokwi](https://wokwi.com/projects/348010712871731794)

- ✓ Run the Python using IDLE to send the Random Values of Level and Weight of the Trash in the bin to IBM Watson IOT platform



The image shows a Python script in a file editor and its execution output in a terminal window. The script, named `1.py`, is located at `C:\Users\Akash M\Desktop\combined code\1.py`. It uses the `wiotp.sdk.device` module to connect to the IBM Watson IoT platform. The script defines a `myCommandCallback` function to handle incoming commands and a `pub` function to publish data. It then enters a `while True` loop that generates random level and weight values and publishes them to the platform. The terminal window shows the output of the script, including connection status, published data, and alerts.

```
1.py - C:\Users\Akash M\Desktop\combined code\1.py (3.7.9)
File Edit Format Run Options Window Help

},
"auth": {
  "token": "1234567890"
}
}

def myCommandCallback (cmd):
    print ("Message received from IBM IoT Platform: %s" % cmd)
    m=cmd.data['command']

client = wiotp.sdk.device.DeviceClient(config=myConfig, lo
client.connect()

def pub (data):
    client.publishEvent(eventId="status", msgFormat="json",
    print ("Published data Successfully: %s", myData)
    if weight == 10:
        print ('ALERT !! Weight is HIGH')
    if level == 10:
        print ('ALERT !! Level is HIGH')

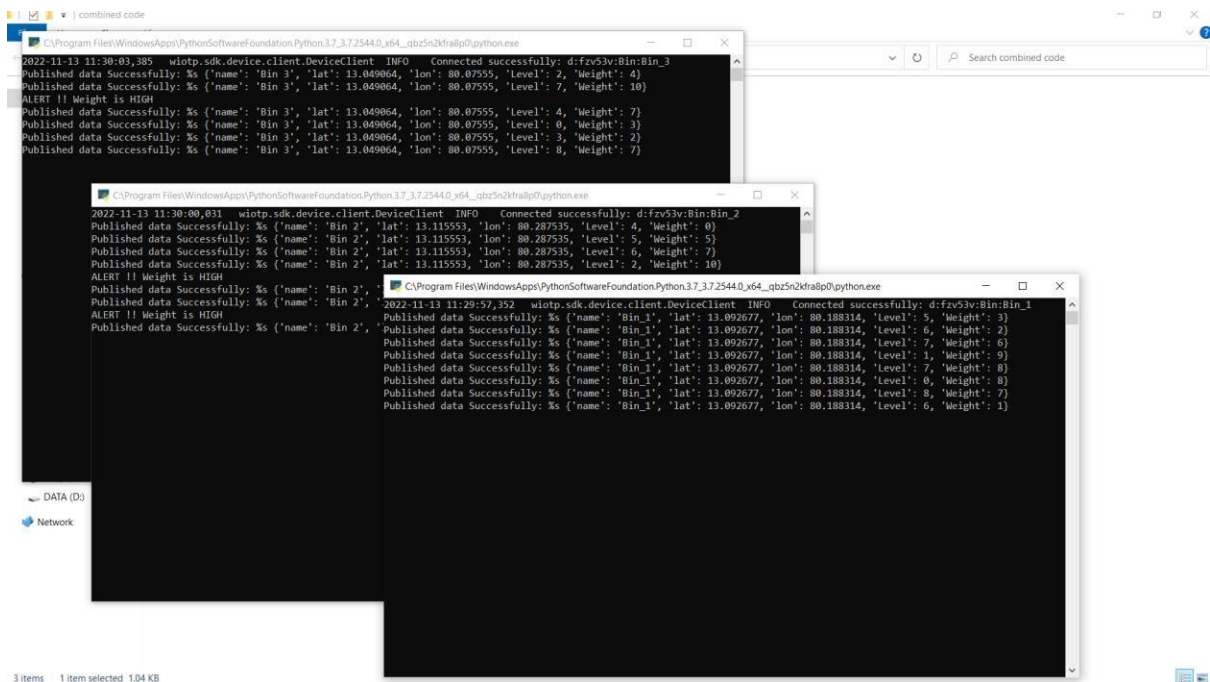
while True:
    level=random.randint(0,10)
    weight=random.randint(0,10)
    myData={'name': 'Bin_1', 'lat': 13.092677, 'lon':
    pub (myData)
    time.sleep (5)
    client.commandCallback = myCommandCallback
client.disconnect ()

Ln:1 Col:0
```

```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 16:30:
00) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for mo
re information.
>>>
===== RESTART: C:\Users\Akash M\Desktop\combined
code\1.py =====
2022-11-13 11:01:22,220 wiotp.sdk.device.client.DeviceC
lient INFO Connected successfully: d:fzv53v:Bin:Bin_1
Published data Successfully: %s {'name': 'Bin_1', 'lat':
13.092677, 'lon': 80.188314, 'Level': 10, 'Weight': 0}
ALERT !! Level is HIGH
Published data Successfully: %s {'name': 'Bin_1', 'lat':
13.092677, 'lon': 80.188314, 'Level': 3, 'Weight': 4}
Published data Successfully: %s {'name': 'Bin_1', 'lat':
13.092677, 'lon': 80.188314, 'Level': 6, 'Weight': 5}
```

## TESTING

- ✓ Executing codes for multiple Bins



The image shows three overlapping terminal windows, each displaying the output of the Python script for a different bin. The windows are titled `combined code`, `combined code`, and `combined code`. Each window shows the connection status, published data, and alerts for a specific bin. The data is as follows:

Bin	Level	Weight
Bin 1	10	0
Bin 1	3	4
Bin 1	6	5
Bin 2	4	0
Bin 2	5	5
Bin 2	6	7
Bin 2	8	10
Bin 3	2	4
Bin 3	7	10
Bin 3	4	7
Bin 3	0	3
Bin 3	3	2
Bin 3	8	7



- ✓ For Example, Three IOT device is create in Watson IOT platform and also connected with the sensor Devices or IDLE . It shows the device is connected

The screenshot displays the IBM Watson IoT Platform interface. At the top, there's a navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons. The main area shows a list of devices under the 'Browse' tab. Three devices are listed: Bin\_1, Bin\_2, and Bin\_3, all with a status of 'Connected'. Bin\_3 is selected, and its details are shown in a modal window. The modal has tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a table of events. The events table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The events are all 'status' events with a 'json' format. A message '1 Simulation running' is visible at the bottom right of the modal.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
Bin_1	Connected	Bin	Device	Nov 10, 2022 6:50 PM	
Bin_2	Connected	Bin	Device	Nov 11, 2022 12:22 PM	
Bin_3	Connected	Bin	Device	Nov 11, 2022 12:25 PM	

Event	Value	Format	Last Received
status	{\"name\":\"Bin 3\",\"lat\":13.049064,\"lon\":80.07555...}	json	a few seconds ago
status	{\"name\":\"Bin 3\",\"lat\":13.049064,\"lon\":80.07555...}	json	a few seconds ago
status	{\"name\":\"Bin 3\",\"lat\":13.049064,\"lon\":80.07555...}	json	a few seconds ago

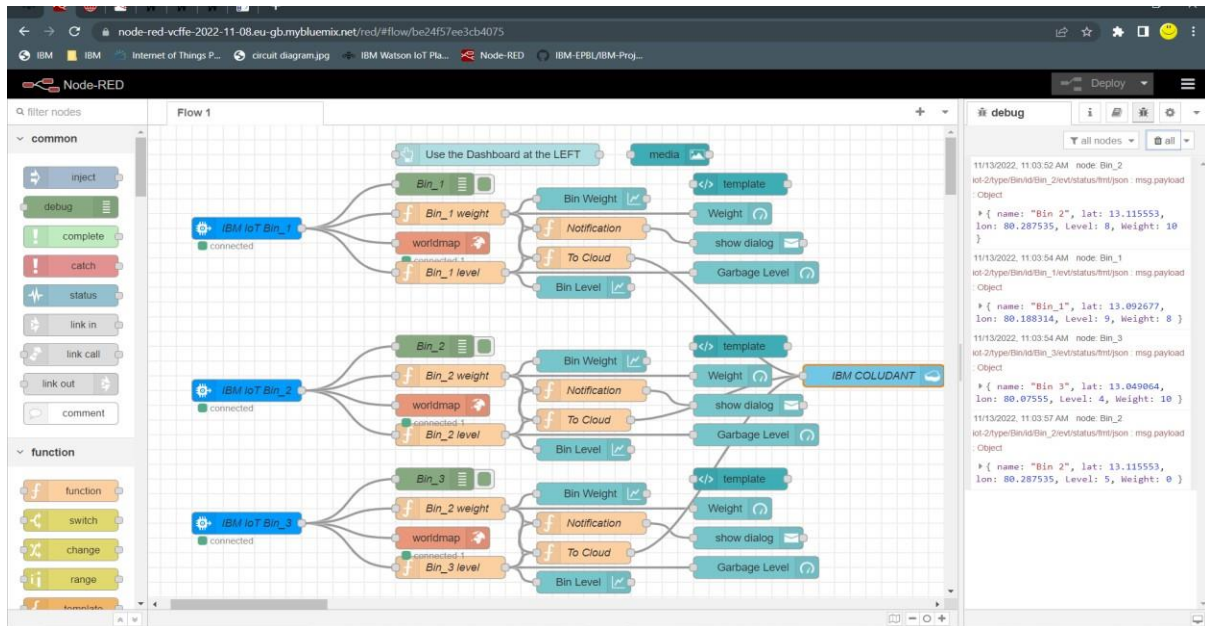
- ✓ The payload received to the IOT device in IBM platform as shown below

The screenshot shows the IBM Watson IoT Platform interface with a modal window titled 'Event Payload' open. The modal displays the event name 'status' and the time received 'Nov 13, 2022 11:04 AM'. The payload is shown as a JSON object: {\"name\": \"Bin 3\", \"lat\": 13.049064, \"lon\": 80.07555, \"Level\": 2, \"Weight\": 7}. The background shows the same device list as the previous screenshot, with Bin\_3 selected.

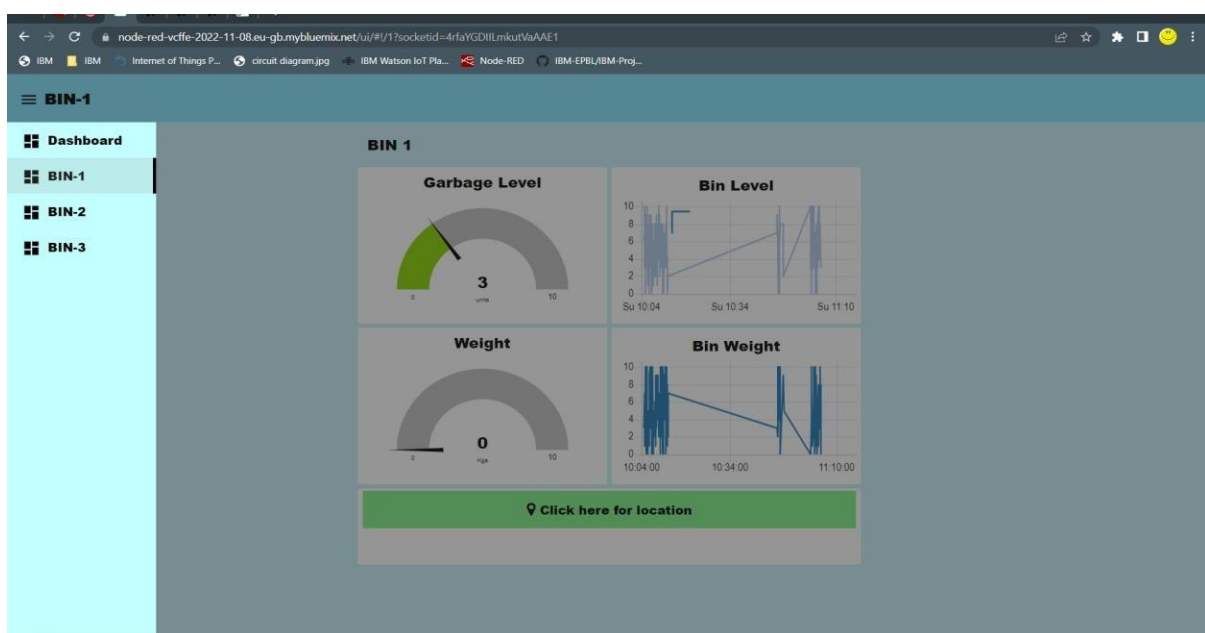
```
1 {
2   \"name\": \"Bin 3\",
3   \"lat\": 13.049064,
4   \"lon\": 80.07555,
5   \"Level\": 2,
6   \"Weight\": 7
7 }
```

- ✓ To Monitor the level and Weight of the bins , the Node-RED platform is used. The Nodes are used to make connections.
- ✓ IBM IOT Node is to receive message from IBM IOT device.
- ✓ Several Functions , Templates , Gauge , Graph Nodes are used to create UI

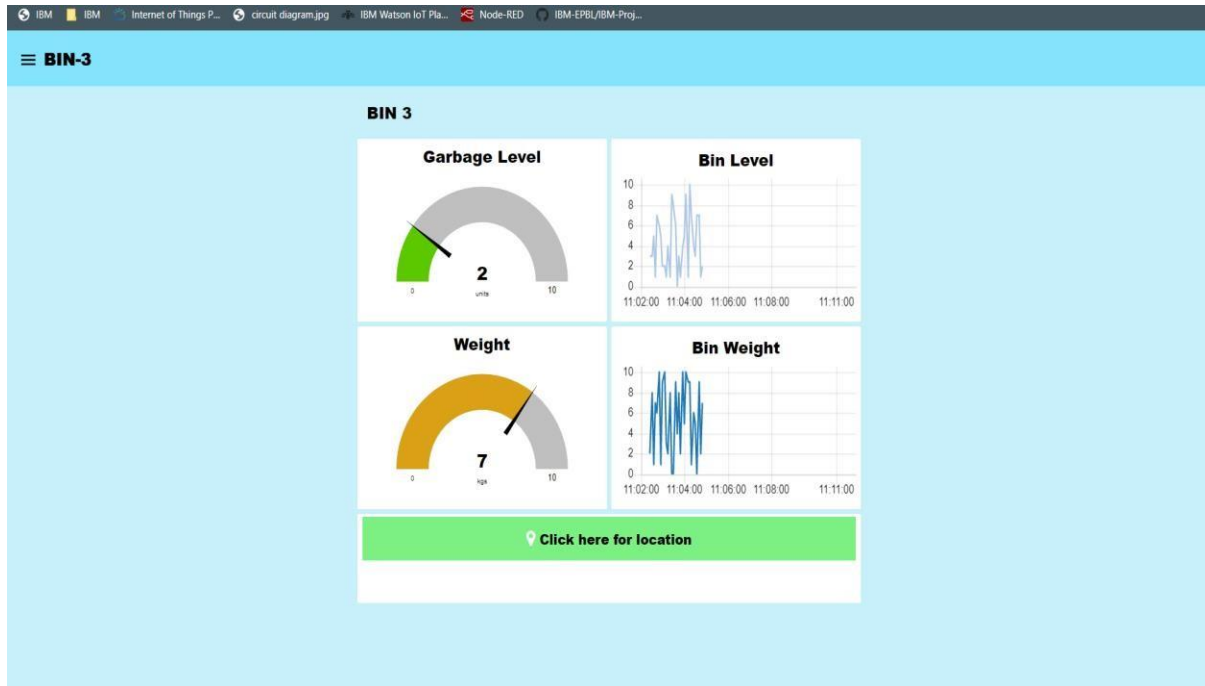
## Web UI : [Smart Waste Management](#)



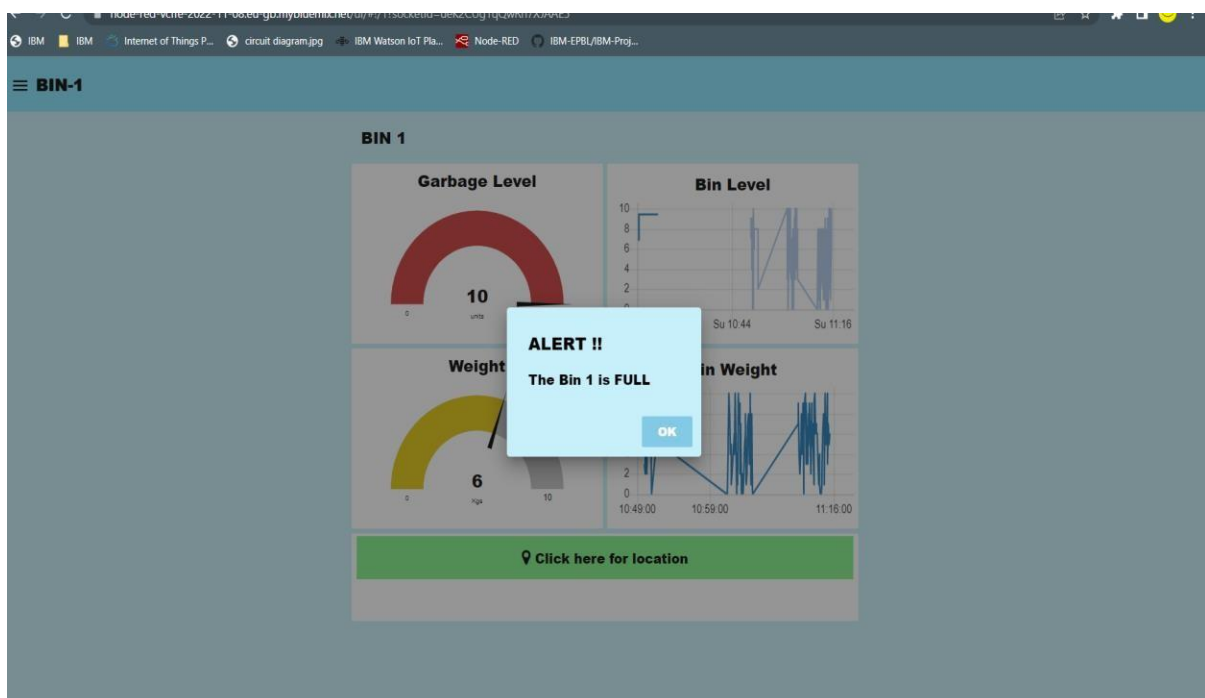
- ✓ The Web UI is created using Node-RED services , The Dashboard is used to Navigate between the Number of Bins



- ✓ The UI is used to monitor the garbage by weight and Level of the Trash as shown Below.
- ✓ The previous Level of the bins can also be viewed using the chart

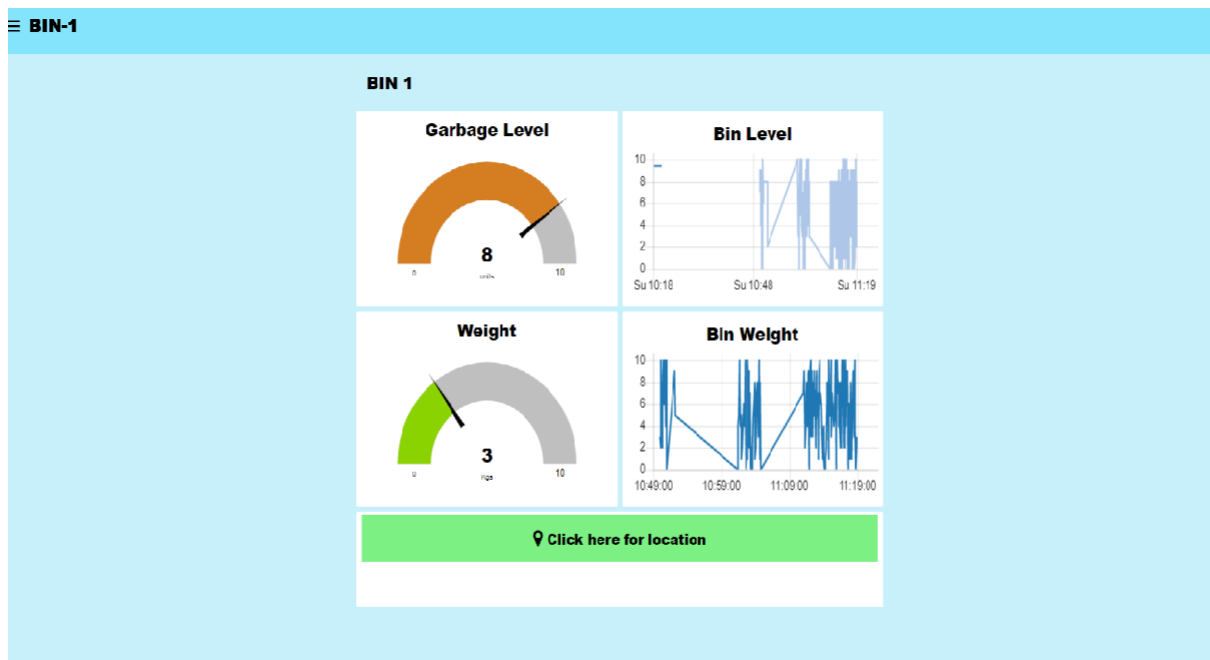


- ✓ If the level of the Trash is FULL or the Weight of the Trash is MAXIMUM , The ALERT message is Displayed

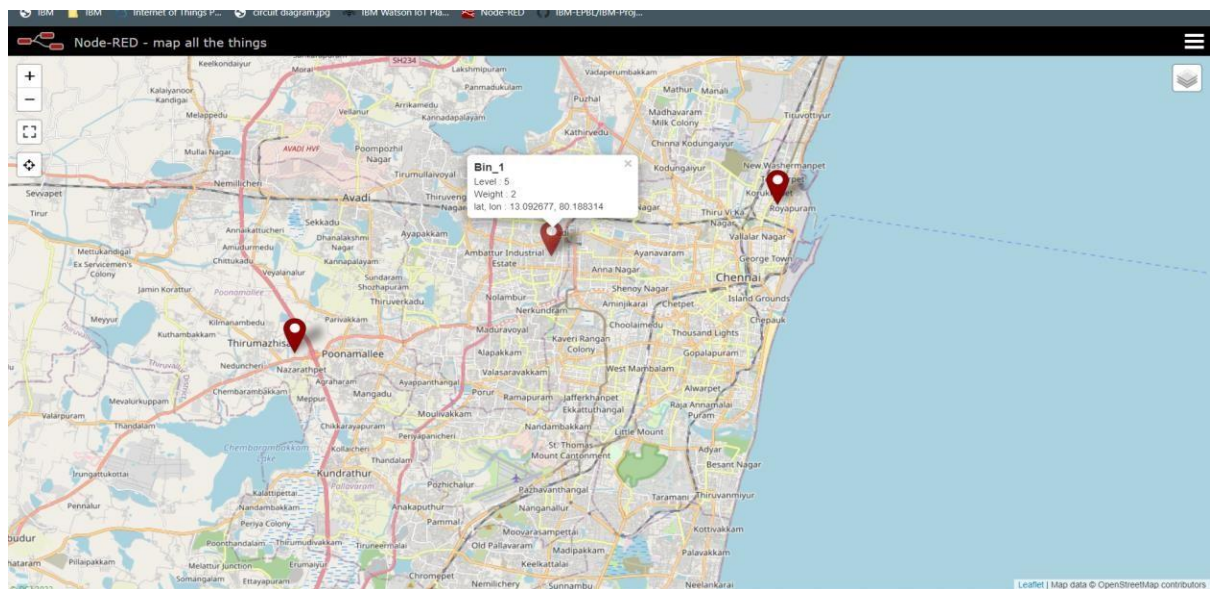




- ✓ The UI also contains the button – “Click here for Location” , which on clicking Navigates to the World map to show the Locations of the Bins

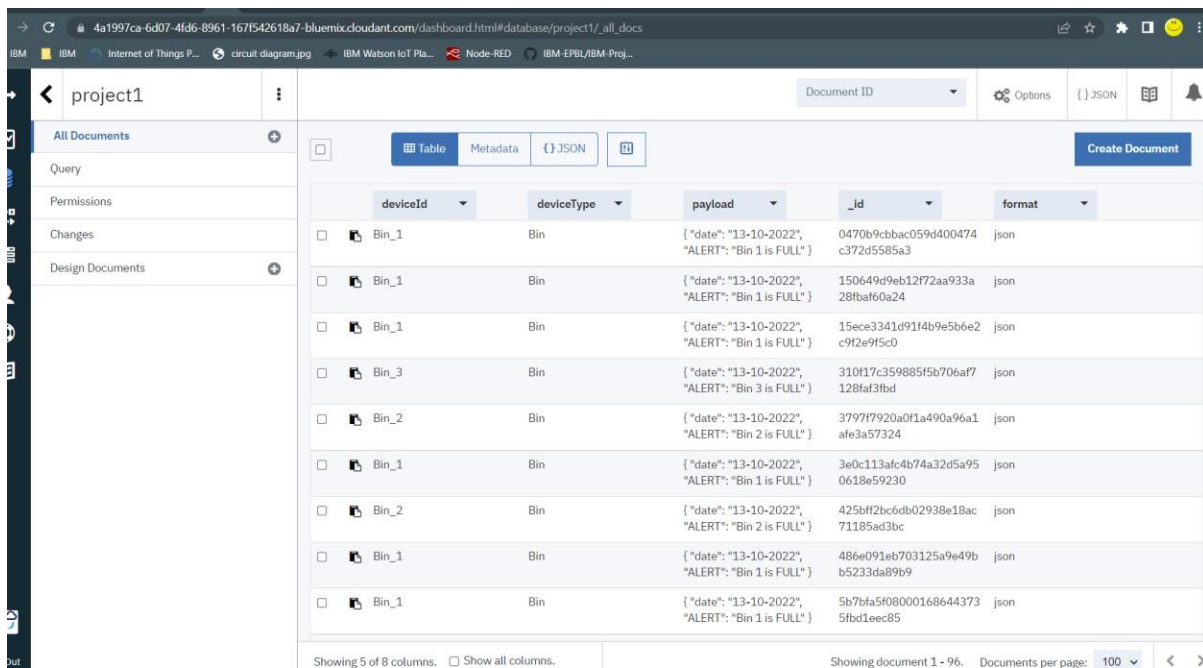


- ✓ The Locations of the Bins is shown on the World map



Location of Bins : [Location](#)

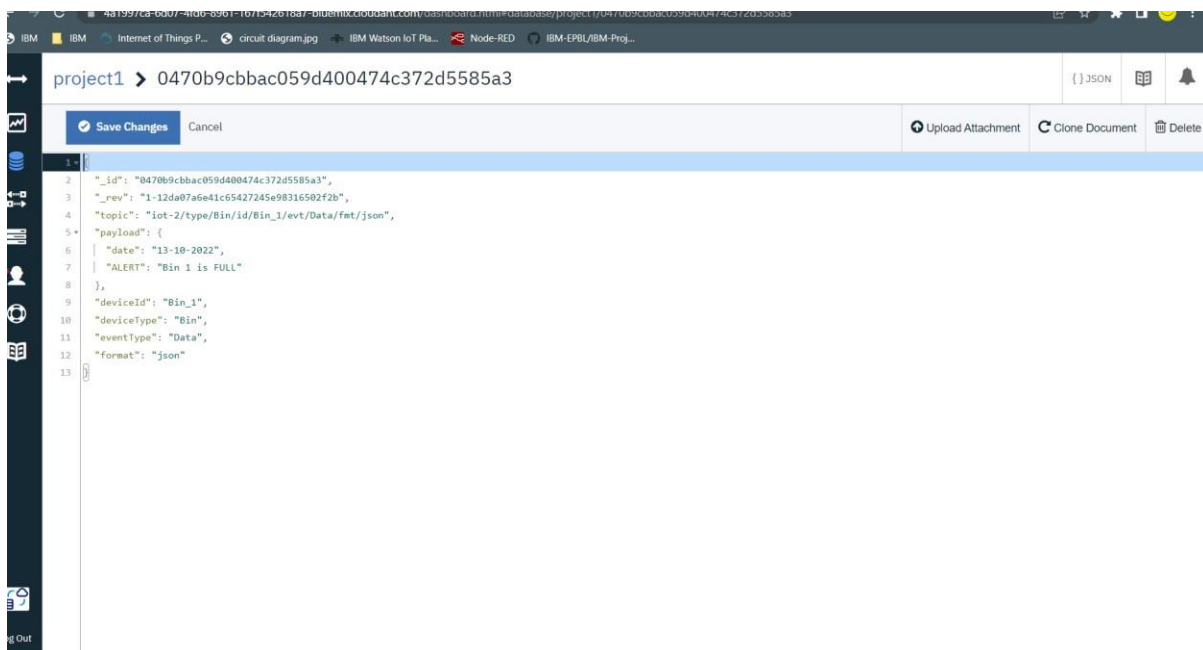
- ✓ When Bin is Full the alert messages are sent and stored to the IBM Cloud which can be viewed if needed



The screenshot shows the IBM Cloud IoT Platform dashboard for 'project1'. The 'Documents' tab is selected, displaying a table of documents. The table has columns for 'deviceId', 'deviceType', 'payload', '\_id', and 'format'. The payload for each document is a JSON object containing a date and an alert message.

deviceId	deviceType	payload	_id	format
Bin_1	Bin	{"date": "13-10-2022", "ALERT": "Bin 1 is FULL"}	0470b9cbbac059d400474c372d5585a3	json
Bin_1	Bin	{"date": "13-10-2022", "ALERT": "Bin 1 is FULL"}	150649d9eb12f72aa933a28fbaf60a24	json
Bin_1	Bin	{"date": "13-10-2022", "ALERT": "Bin 1 is FULL"}	15ece3341d91f4b9e5b6e2c9f2e9f5c0	json
Bin_3	Bin	{"date": "13-10-2022", "ALERT": "Bin 3 is FULL"}	310f17c359885f5b706af7128faf3fbd	json
Bin_2	Bin	{"date": "13-10-2022", "ALERT": "Bin 2 is FULL"}	3797f7920a0f1a490a96a1afe3a57324	json
Bin_1	Bin	{"date": "13-10-2022", "ALERT": "Bin 1 is FULL"}	3e0c113afc4b74a32d5a950618e59230	json
Bin_2	Bin	{"date": "13-10-2022", "ALERT": "Bin 2 is FULL"}	425bf12bc6db02938e18ac71185ad3bc	json
Bin_1	Bin	{"date": "13-10-2022", "ALERT": "Bin 1 is FULL"}	486e091eb703125a9e49bb5233da89b9	json
Bin_1	Bin	{"date": "13-10-2022", "ALERT": "Bin 1 is FULL"}	5b7bfa5f080001686443735fbd1eec85	json

- ✓ The Message stored in cloud contains Bin details , Date and The ALERT message



The screenshot shows the details of a specific document in the IBM Cloud IoT Platform dashboard. The document ID is '0470b9cbbac059d400474c372d5585a3'. The document is a JSON object containing the following fields:

```
{
  "_id": "0470b9cbbac059d400474c372d5585a3",
  "_rev": "1-12da07a6e1c65427245e98316502f2b",
  "topic": "Iot-2/type/Bin/id/Bin_1/evt/Data/fmt/json",
  "payload": {
    "date": "13-10-2022",
    "ALERT": "Bin 1 is FULL"
  },
  "deviceId": "Bin_1",
  "deviceType": "Bin",
  "eventType": "Data",
  "format": "json"
}
```