

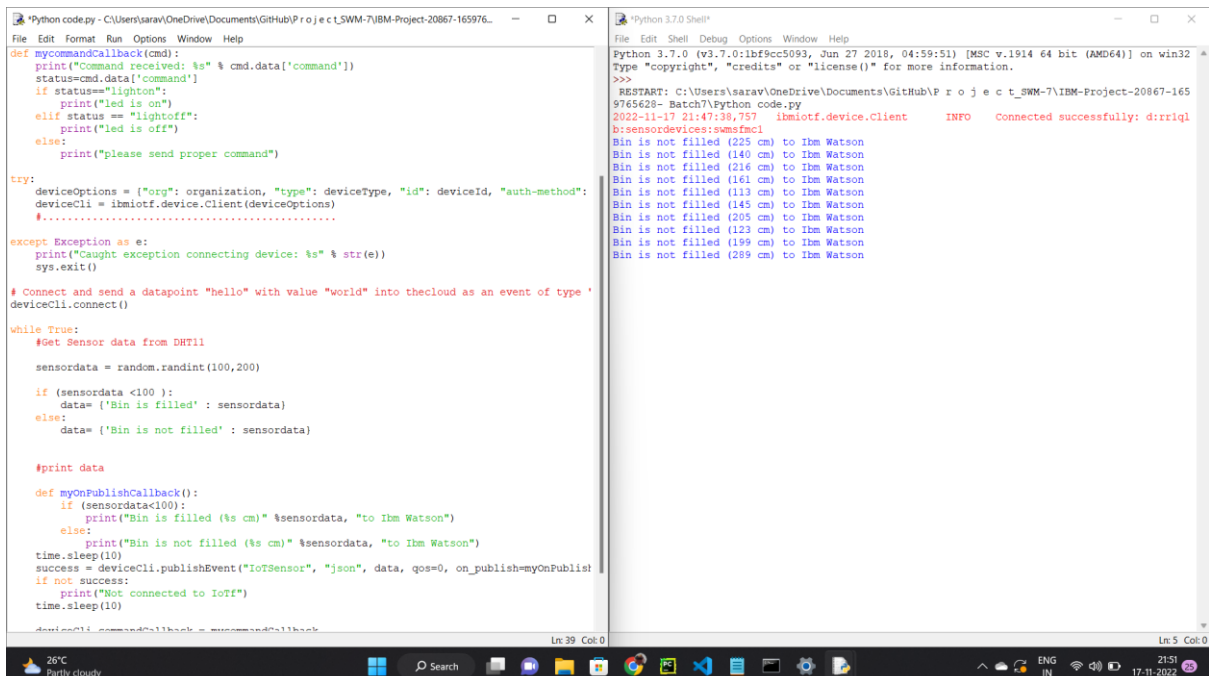
Sprint 2

| | |
|--------------|---|
| Date | 10-11-2022 |
| Team Id | PNT2022TMID22482 |
| Project Name | Project -Smart Waste Management System for Metropolitan Cities |
| Team Members | Saravanan G Anand Krishnan N Kishore Kumar J Mohamed Idris M |

Description:

- A device is created in IBM IoT Watson platform and simulated.
- Node-red flow and dashboard is created to show the data from the cloud
- A python script is written to send the data to IOT platform and node-red dashboard from the code.

Python running module:



```
Python code.py - C:\Users\sarav\OneDrive\Documents\GitHub\Project_SWM-7\IBM-Project-20867-165976...
File Edit Format Run Options Window Help
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="lighton":
        print("led is on")
    elif status == "lightoff":
        print("led is off")
    else:
        print("please send proper command")

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #.....

except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

# Connect and send a datapoint "hello" with value "world" into thecloud as an event of type '
deviceCli.connect()

while True:
    #Get Sensor data from DHT11
    sensordata = random.randint(100,200)

    if (sensordata <100 ):
        data= {'Bin is filled' : sensordata}
    else:
        data= {'Bin is not filled' : sensordata}

    #print data
    def myOnPublishCallback():
        if (sensordata<100):
            print("Bin is filled (%s cm)" %sensordata, "to Ibm Watson")
        else:
            print("Bin is not filled (%s cm)" %sensordata, "to Ibm Watson")
    time.sleep(10)
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublish
    if not success:
        print("Not connected to IoTf")
    time.sleep(10)

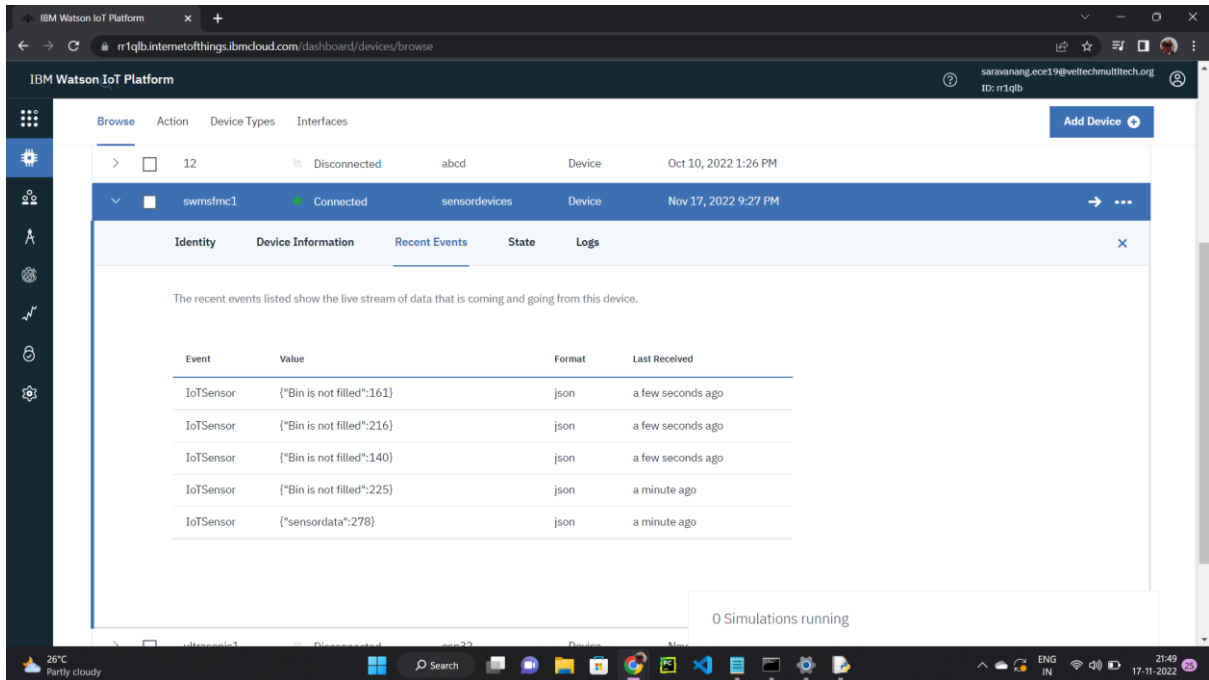
deviceCli.commandCallback = myCommandCallback

Ln: 39 Col: 0

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\sarav\OneDrive\Documents\GitHub\Project_SWM-7\IBM-Project-20867-165
9765628- Batch7\Python code.py
2022-11-17 21:47:38,757 - ibmiotf.device.Client - INFO - Connected successfully: d:rrlql
b:sensordata:sensordata
Bin is not filled (225 cm) to Ibm Watson
Bin is not filled (140 cm) to Ibm Watson
Bin is not filled (216 cm) to Ibm Watson
Bin is not filled (161 cm) to Ibm Watson
Bin is not filled (113 cm) to Ibm Watson
Bin is not filled (145 cm) to Ibm Watson
Bin is not filled (205 cm) to Ibm Watson
Bin is not filled (123 cm) to Ibm Watson
Bin is not filled (199 cm) to Ibm Watson
Bin is not filled (289 cm) to Ibm Watson

Ln: 5 Col: 0
```

Data sent from Module to cloud:

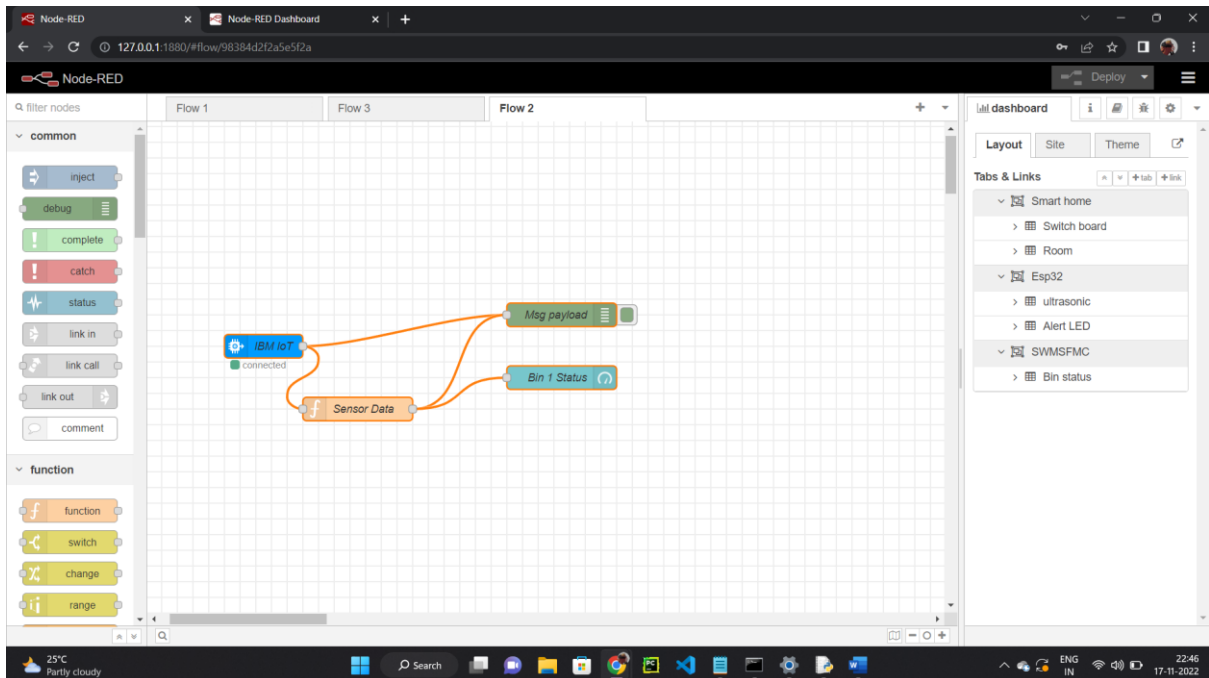


The screenshot shows the IBM Watson IoT Platform dashboard. The device 'swmsfmc1' is connected. The 'Recent Events' tab is active, displaying a table of events received from the device.

| Event | Value | Format | Last Received |
|-----------|---------------------------|--------|-------------------|
| IoTSensor | {"Bin is not filled":161} | json | a few seconds ago |
| IoTSensor | {"Bin is not filled":216} | json | a few seconds ago |
| IoTSensor | {"Bin is not filled":140} | json | a few seconds ago |
| IoTSensor | {"Bin is not filled":225} | json | a minute ago |
| IoTSensor | {"sensordata":278} | json | a minute ago |

0 Simulations running

Node-red flow:



The screenshot shows the Node-RED flow editor. The flow is titled 'Flow 2' and is currently active. It consists of the following nodes and connections:

- IBM IoT** node (connected) is connected to the **Sensor Data** function node.
- The **Sensor Data** function node is connected to the **Msg payload** node.
- The **Msg payload** node is connected to the **Bin 1 Status** node.

The left sidebar shows the 'common' and 'function' node palettes. The right sidebar shows the 'dashboard' settings and 'Tabs & Links' section.

Node-red Dashboard:

