

SPRINT - 3

Date	12 NOV 2022
Team ID	PNT2022TMID01880
Project Name	Smart Waste Management System for Metropolitan Cities

1, Simulate python code in Python IDE software to transmit data to IBM Watson IOT platform

Python code:

smartbin.py:

#Project: Smart Waste Management System for Metropolitan cities #Team ID: PNT2022TMID01880

#Installing necessary libraries

import wiotp.sdk.device

import time

import random

import requests

import math

#Configuration details for connecting python script to IBM Watson IOT Platform

myConfig = { "identity": {

"orgId": "1hx03x",

"typeId": "cloud",

"deviceId": "232323"

},

"auth": {

"token": "12345678"

}}

def myCommandCallback(cmd):

print("Message received from IBM IoT Platform: %s" % cmd.data)

#Connecting the client to ibm watson iot platform

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

```
client.commandCallback = myCommandCallback
client.connect()
print("CONNECTED");
```

```
while True:
```

```
    res = requests.get('https://ipinfo.io/')
    data = res.json()
    loc = data['loc'].split(',')
    theta = random.uniform(0,2*math.pi)
    area = (0.05**2)*math.pi
    radius = math.sqrt(random.uniform(0,area/math.pi))
    latitude,longitude = [float(loc[0])+radius*math.cos(theta), float(loc[1])+radius*math.sin(theta)]

    binlevel=random.randint(10,100)
    distance = random.randint(10,100)
    if binlevel>80:
        myData={'Distance':distance, 'latitude':latitude, 'longitude':longitude,'binlevel':binlevel}
        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)
        client.commandCallback = myCommandCallback
        print("BIN IS FULL...")
        print("SENDING THE DATA...")
        time.sleep(2)
    else :
        print("BIN IS IN NORMAL LEVEL...")
        time.sleep(2)

#Disconnect the client connection
client.disconnect()
```

Python IDE output:

#Project: Smart Waste Management System for Metropolitan cities #Team ID: PNT2022TMID01880

```
import math
```

1. *Journal of the American Medical Association*, 1997; 277: 1039-1043.

```
print("Message received from IBM IoT Platform: %s" % cmd.data)
```

```
print ("CONNECTED");
```

```
distance = rand(
```

```
client.publishEvent(eventId="status" msgFormat="json"
```

BIN IS IN NORMAL LEVEL...

Ln: 48 Col: 39

2. Data is transferred to IBM Watson IoT platform.

IBM Platform output:

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes links for Browse, Action, Device Types, and Interfaces. A sidebar on the left contains icons for various platform features. The main content area displays a table of devices, with one device selected for detailed viewing.

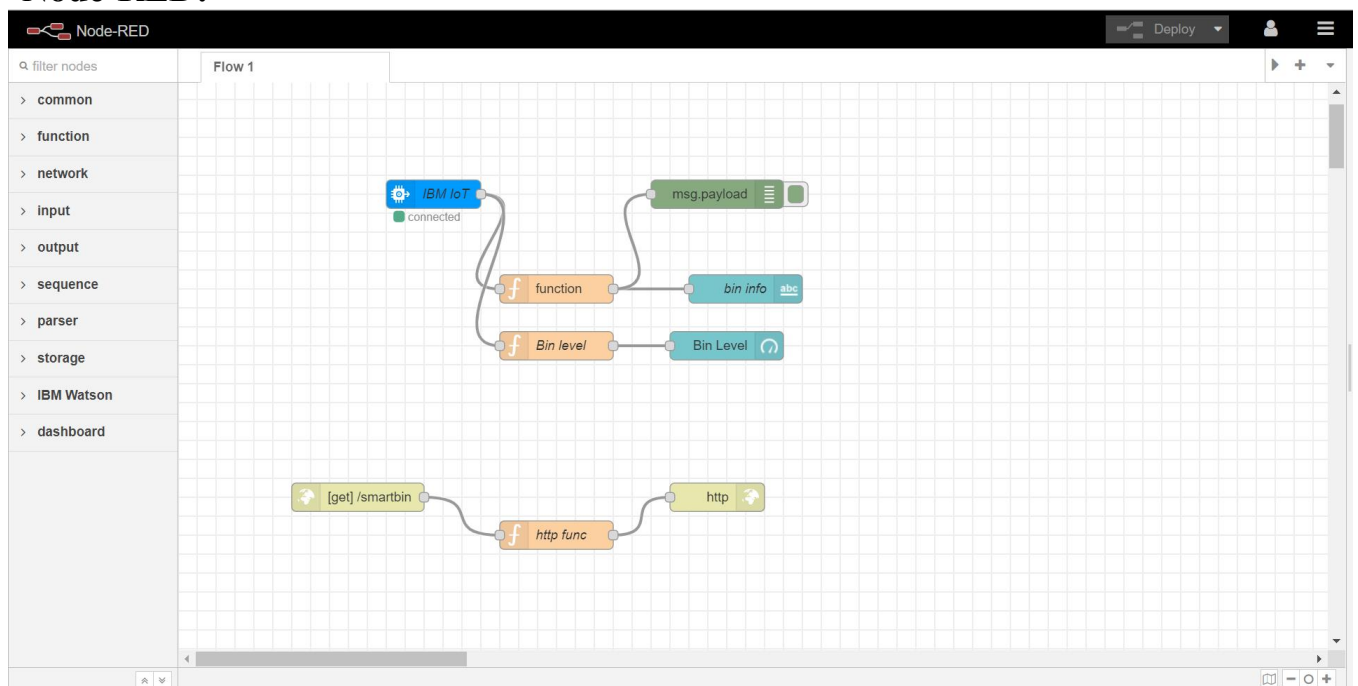
Device ID: 232323, Status: Connected, Device Type: cloud, Class ID: Device, Date Added: Nov 20, 2022 12:23 PM, Descriptive Location: 7376191ec309@smartintmz.com, Added By: 7376191ec309@smartintmz.com, Device Class: ...

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
status	("Distance":99,"latitude":11.237919061707885...	json	a few seconds ago
status	("Distance":75,"latitude":11.21336255027092,"...	json	a few seconds ago
status	("Distance":61,"latitude":11.225098748763434...	json	23 minutes ago
status	("Distance":66,"latitude":11.215185174421823...	json	23 minutes ago
status	("Distance":41,"latitude":11.206082076734994...	json	23 minutes ago

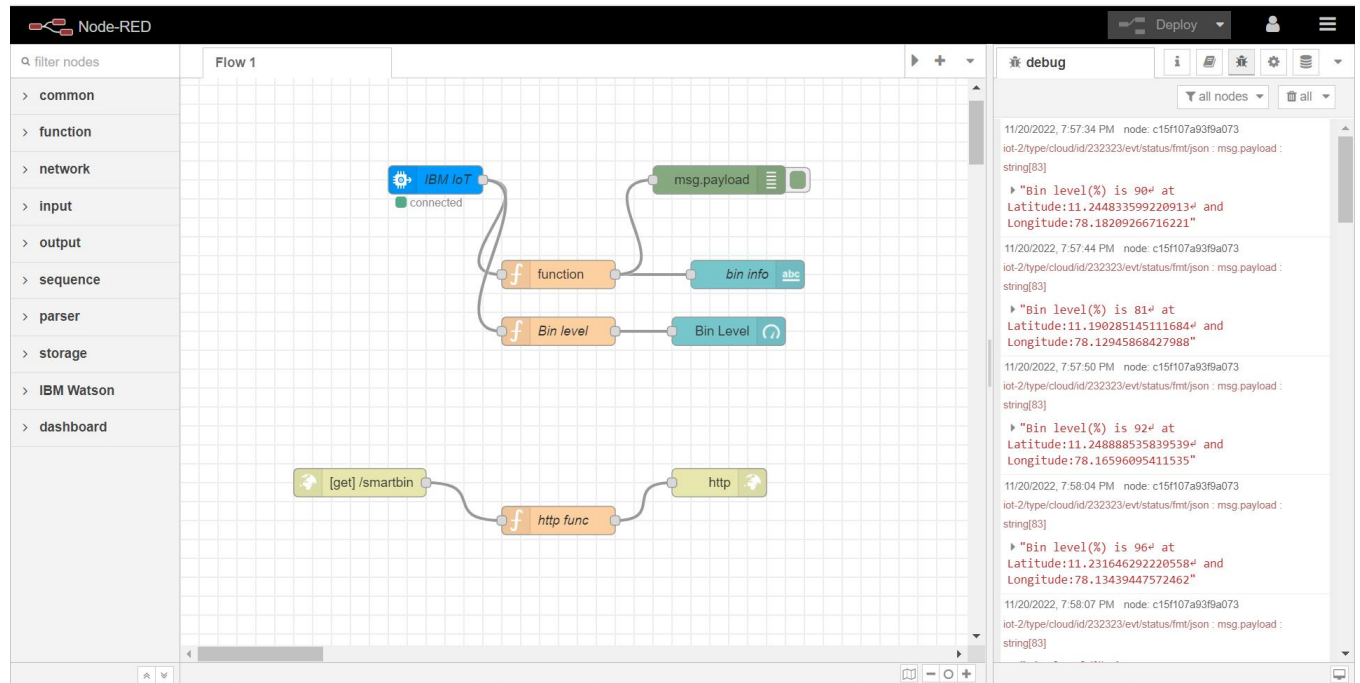
3. Data transfer from IBM Watson IOT platform and Python IDE to Node RED.

Node-RED:



4. Node-RED Connection setup for data transmission from IBM Watson IoT platform to Node-RED dashboard and viewing in Web UI .

Node-RED:



Web UI:

