**Real-Time River Water Quality Monitoring and Control Systems**

**Team ID – PNT2022TMID07793**

**Develop a python script and upload to ibm cloud**

**Code:**

import random

import time import sys

import ibmiotf.application

import ibmiotf.device

# Provide your IBM Watson Device Credentials

organization = "uwujz1"

deviceType = "ibm\_iot" deviceId = "Python\_iot" authMethod = "token"

authToken = "1234asdf"

# repalce it with organization ID

# replace it with device type # repalce with device id

# repalce with token

def myCommandCallback(cmd):

print("Command received: %s" % cmd.data) if cmd.data['command'] == 'lighton':

print("LIGHT ON")

elif cmd.data['command'] == 'lightoff': print("LIGHT OFF")

try:

deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod,

"auth-token": authToken} deviceCli = ibmiotf.device.Client(deviceOptions)

# ..............................................

except Exception as e:

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

deviceCli.connect()

while True:

pH = random.randint(0,100) conductivity = random.randint(0,100) T = random.randint(0,100)

oxygen = random.randint(0,100) turbidity = random.randint(0,100)

# Send Temperature & Humidity to IBM Watson data = {"turbidity":turbidity'temp':

T,'ph':pH,'Salinity':conductivity,'oxygen':oxygen}

# print data

def myOnPublishCallback():

print("Published data",data, "to IBM Watson")

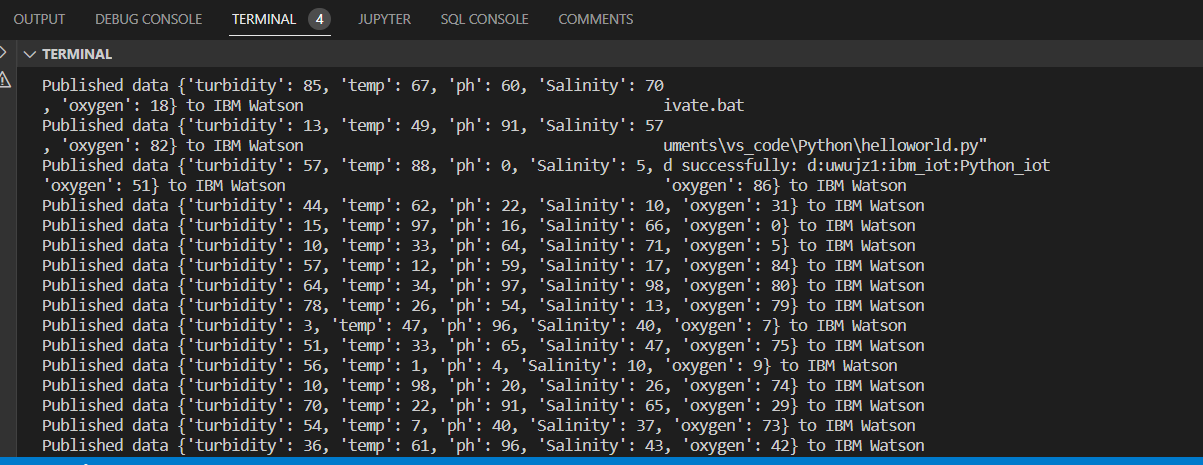
success = deviceCli.publishEvent("event", "json", data, 0, myOnPublishCallback)

if not success:

print("Not connected to IoTF") time.sleep(5)

deviceCli.commandCallback = myCommandCallback

**Output :**



**Ibm cloud output:**

