

<b>TEAM ID</b>	<b>PNT2022TMID00945</b>
<b>PROJECT NAME</b>	<b>GAS LEAKAGE MONITORING AND ALERTING SYSTEM</b>

## **PYTHON CODE TO PUBLISH DATA TO IBM CLOUD**

### **Python Code:**

#IBM Watson IOT Platform

#pip install wiotp-sdk

import wiotp.sdk.device

import time import random

myConfig = {

    "identity": {

        "orgId": "0tus0f",

        "typeId": "ESP32",

        "deviceId": "01"

    },

    "auth": {

        "token": "Gowth@m@nk18"

    }

}

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() while True:

    temp=random.randint(-20,125)

```
hum=random.randint(0,100)

myData={'temperature':temp, 'humidity':hum}

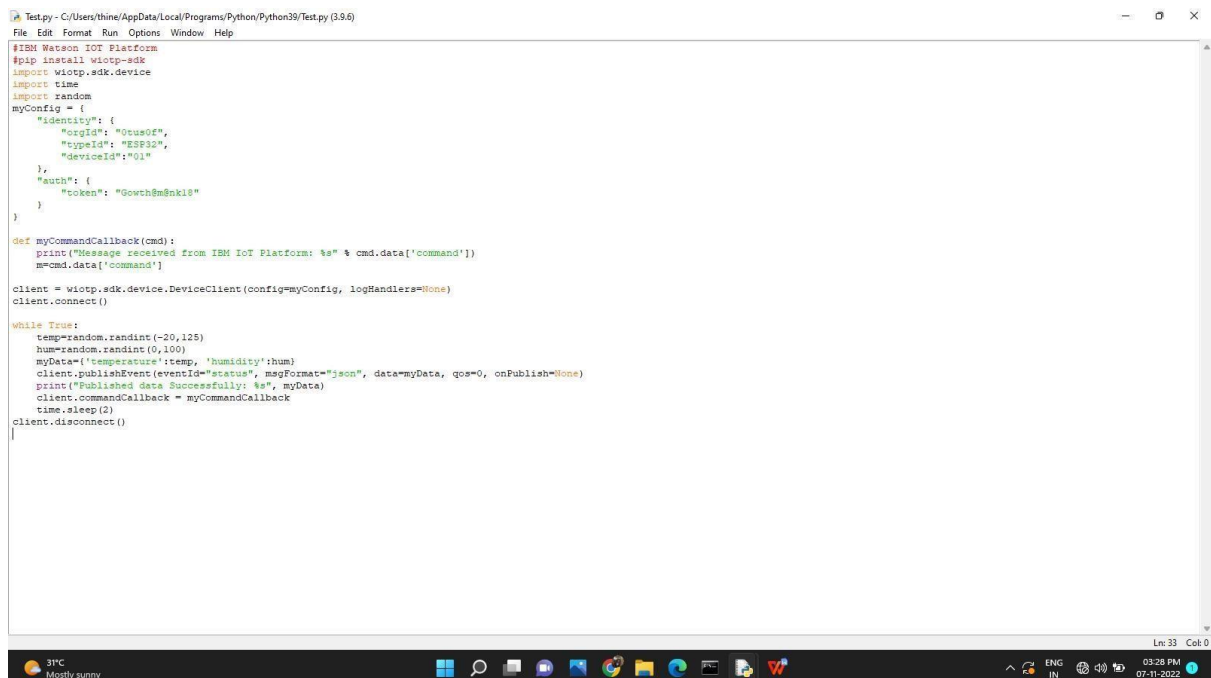
client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)

print("Published data Successfully: %s", myData)

client.commandCallback = myCommandCallback time.sleep(2)

client.disconnect()
```

## PYTHON CODE:

A screenshot of a Windows desktop environment. At the top, a text editor window titled 'Test.py - C:/Users/thine/AppData/Local/Programs/Python/Python39/Python39/Test.py (3.9.6)' is open. The editor contains a Python script for connecting to the IBM Watson IoT Platform. The script imports necessary modules, defines a configuration dictionary, sets up a command callback, connects to the platform, and enters a loop that publishes temperature and humidity data every 2 seconds. The Windows taskbar at the bottom shows the system clock as 03:28 PM on 07-11-2022, along with various system icons and the language set to ENG IN.

```
Test.py - C:/Users/thine/AppData/Local/Programs/Python/Python39/Python39/Test.py (3.9.6)
File Edit Format Run Options Window Help

#IBM Watson IoT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "0tuse0f",
        "typeId": "ESP32",
        "deviceId": "01"
    },
    "auth": {
        "token": "Gowth@m$nk18"
    }
}

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()

while True:
    temp=random.randint(-20,125)
    hum=random.randint(0,100)
    myData={'temperature':temp, 'humidity':hum}
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

## PYTHON OUTPUT:

```
Test.py - C:/Users/thine/AppData/Local/Programs/Python/Python39/Test.py (3.9.6)
File Edit Shell Debug Options Window Help
#IDE
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/thine/AppData/Local/Programs/Python/Python39/Test.py =====
2022-11-07 16:28:59,770 wiotp.sdk.device.client.DeviceClient INFO Connecte
d successfully: d:\otus0f:ESP32:01
Published data Successfully: %s ('temperature': 27, 'humidity': 26)
Published data Successfully: %s ('temperature': 100, 'humidity': 75)
Published data Successfully: %s ('temperature': 115, 'humidity': 69)
Published data Successfully: %s ('temperature': 18, 'humidity': 69)
Published data Successfully: %s ('temperature': 77, 'humidity': 40)
Published data Successfully: %s ('temperature': 12, 'humidity': 72)
Published data Successfully: %s ('temperature': 117, 'humidity': 33)
Published data Successfully: %s ('temperature': 32, 'humidity': 15)
Published data Successfully: %s ('temperature': 25, 'humidity': 19)
Published data Successfully: %s ('temperature': 124, 'humidity': 26)
|
c114
c114
while
c114
sh=None)
c114
Ln: 33 Col: 0
```

## Watson Cloud IBM:

Watson Cloud IBM interface showing device management and data stream.

Search by Device ID:

Device Simulator: ☒

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By
01	Connected	ESP32	Device	Nov 6, 2022 9:53 AM		710019106014@smartinternz.com

Recent Events

Event	Value	Format	Last Received
status	["temperature":46,"humidity":68]	json	a few seconds ago
status	["temperature":33,"humidity":16]	json	a few seconds ago
status	["temperature":116,"humidity":4]	json	a few seconds ago
status	["temperature":46,"humidity":1]	json	a few seconds ago
status	["temperature":116,"humidity":32]	json	a few seconds ago

0 Simulations running