

DEVELOP A PYTHON SCRIPT TO PUBLISH AND SUBSCRIBE TO IBM IOT PLATFORM

DATE	6 November 2022
TEAM ID	PNT2022TMID30897
PROJECT TITTLE	Gas Leakage Monitoring and Alerting System

Develop python code :

```
import time
import sys

import ibmiotf.application
import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "u9pz01" deviceType = "abcd"

deviceId = "temphum" authMethod = "token"

authToken = "12345678"

# Initialize GPIO

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")

        #print(cmd)

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()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type
"greeting" 10 times deviceCli.connect()

while True:
        #Get Sensor Data from DHT11

        temp=random.randint(90,110)
        Humid=random.randint(60,100)

        data = { 'temp' : temp, 'Humid': Humid }

        #print data      def
myOnPublishCallback():

```

```
print ("Published Temperature = %s C" % temp, "Humidity = %s %" % Humid, "to  
IBM Watson")
```

```
success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,  
on_publish=myOnPublishCallback)
```

```
if not success:
```

```
print("Not connected to IoT")
```

```
time.sleep(10)
```

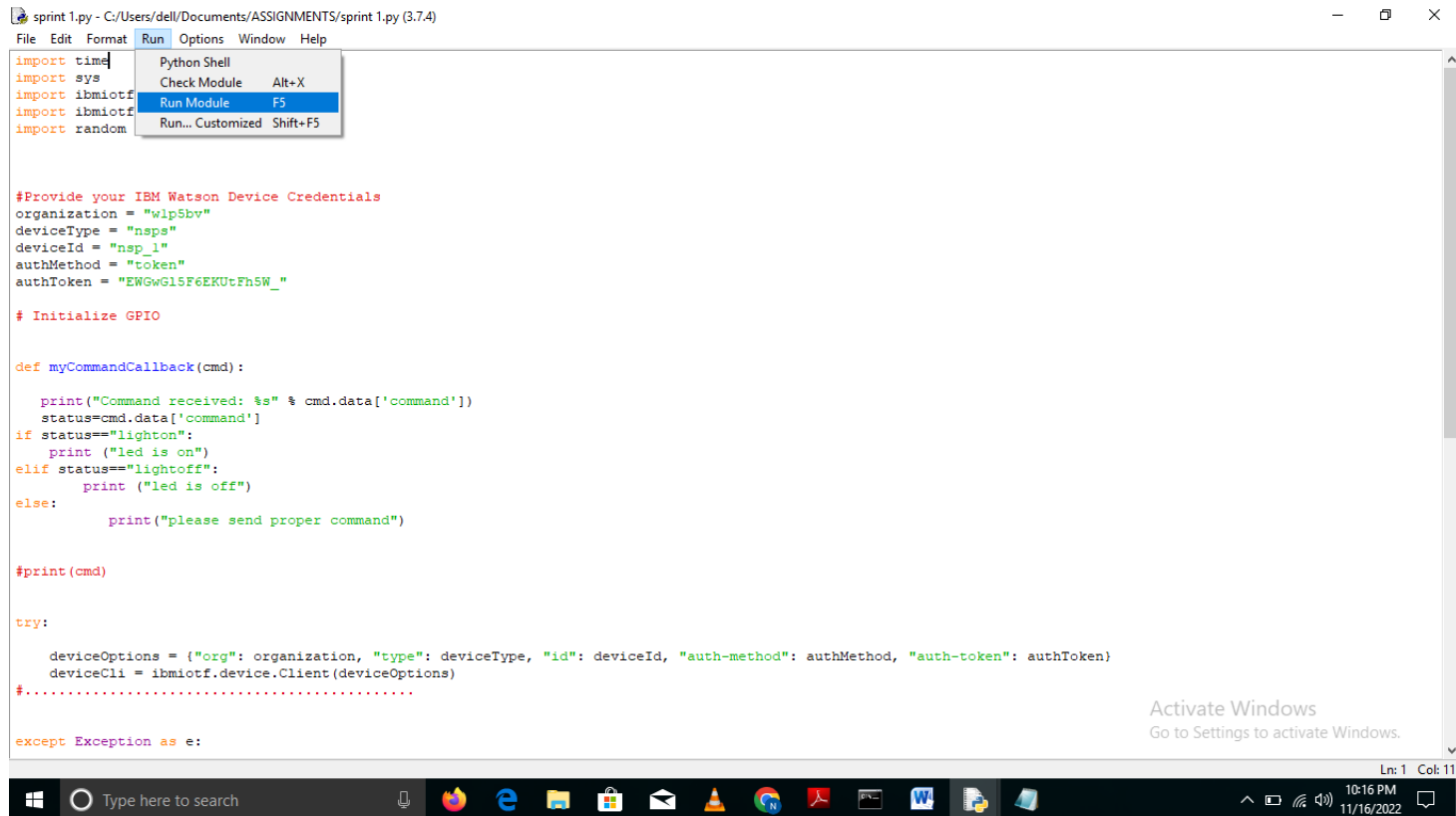
```
deviceCli.commandCallback = myCommandCallback
```

```
# Disconnect the device and application from the cloud deviceCli.disconnect()
```

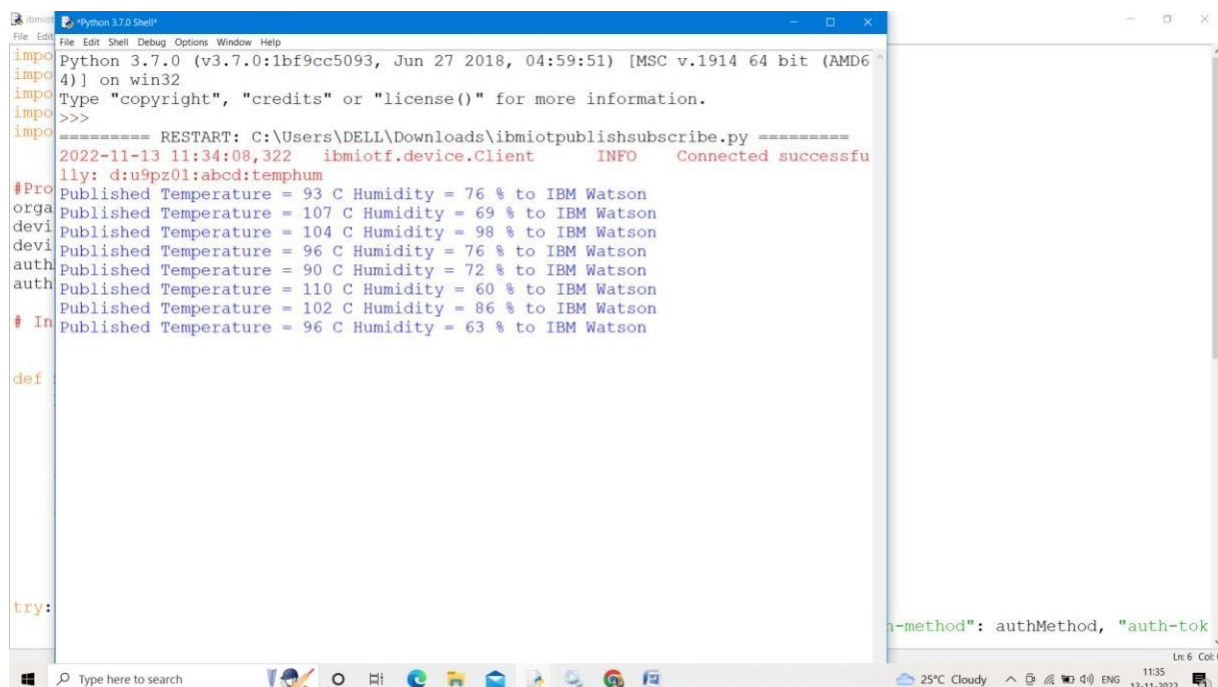
Publish data to IBM Cloud:

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes the IBM logo, the text "IBM Watson IoT Platform", and a user profile section for "sriramnathiya@gmail.com" with ID "w1p5bv". The main content area is titled "Browse" and contains a table of devices. The table has columns for Device ID, Status, Device Type, Class ID, and Date Added. There are four devices listed, all with a status of "Disconnected". A search bar and a "Device Simulator" toggle are also visible. The bottom of the screen shows a Windows taskbar with various application icons and a system clock indicating 10:14 PM on 11/16/2022.

Device ID	Status	Device Type	Class ID	Date Added
nathi_1	Disconnected	nathi	Device	Nov 14, 2022 10:46 PM
nathi_2	Disconnected	nathi	Device	Nov 15, 2022 9:21 PM
nsp_1	Disconnected	nsps	Device	Nov 13, 2022 3:10 PM
sdje_2	Disconnected	sdje	Device	Nov 12, 2022 10:50 PM



Code Output:



IBM Watson Output:

The screenshot displays the IBM Watson IoT Platform web interface. The main dashboard shows a list of devices, with 'nathi_2' selected and its status 'Disconnected'. A modal window is open for configuring 'event_1' for the device type 'nathi'. The modal includes a 'Schedule' section set to 'Every Minute' and a 'Payload' section with a JSON template for temperature, humidity, and gas data. The background interface shows a table of recent events for 'nathi_2'.

Device Type: nathi

Events 1 [New event type +](#)

Event type name: [Send](#) [Delete](#)

Schedule

Payload

Specify the event payload in the editor window or by uploading a [CSV file](#).

```
0 {
1   "temperature": random(0, 100),
2   "humidity": random(0, 100),
3   "gas": random(0, 100)
4 }
5
```

[Upload a CSV file](#)

Recent Events

Event	Value
event_1	{"temp":82,"hum":54,"gas":42}
event_1	{"temp":6,"hum":44,"gas":89}
event_1	{"temp":48,"hum":49,"gas":51}
event_1	{"temp":53,"hum":29,"gas":79}

Activate Windows. Go to Settings to activate Windows.

