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

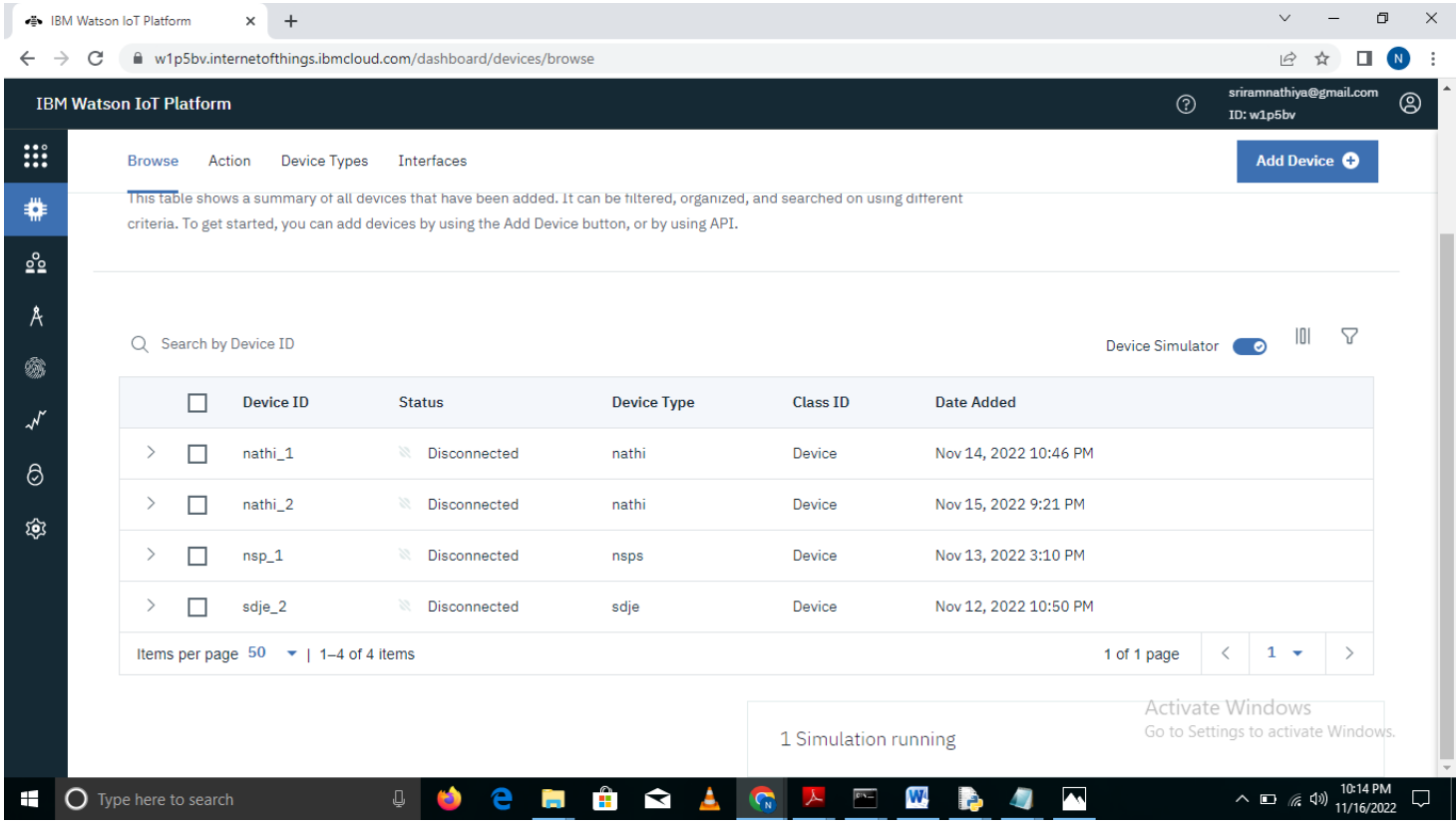
Develop python code :

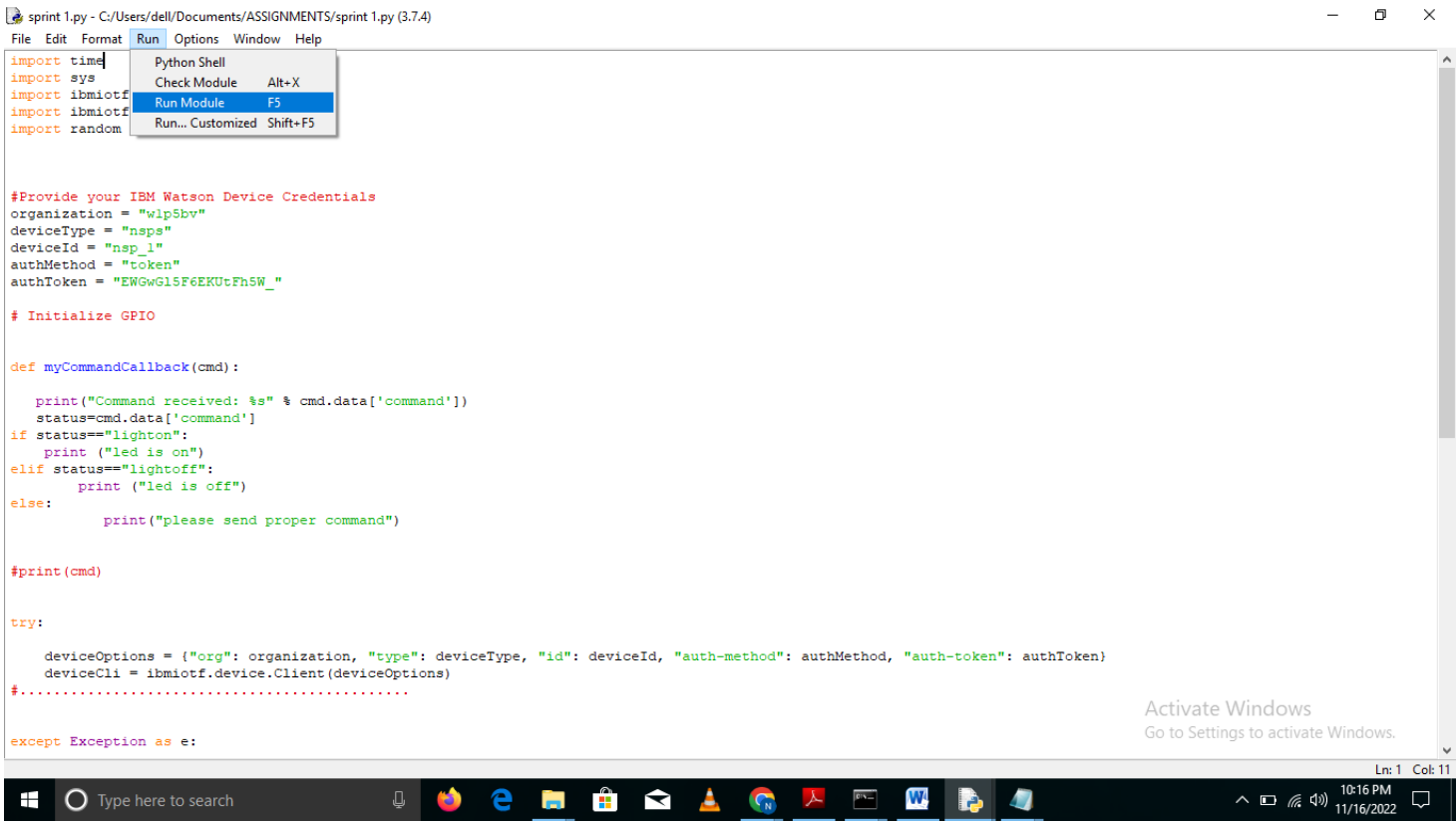
```
import time import sys

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():
```

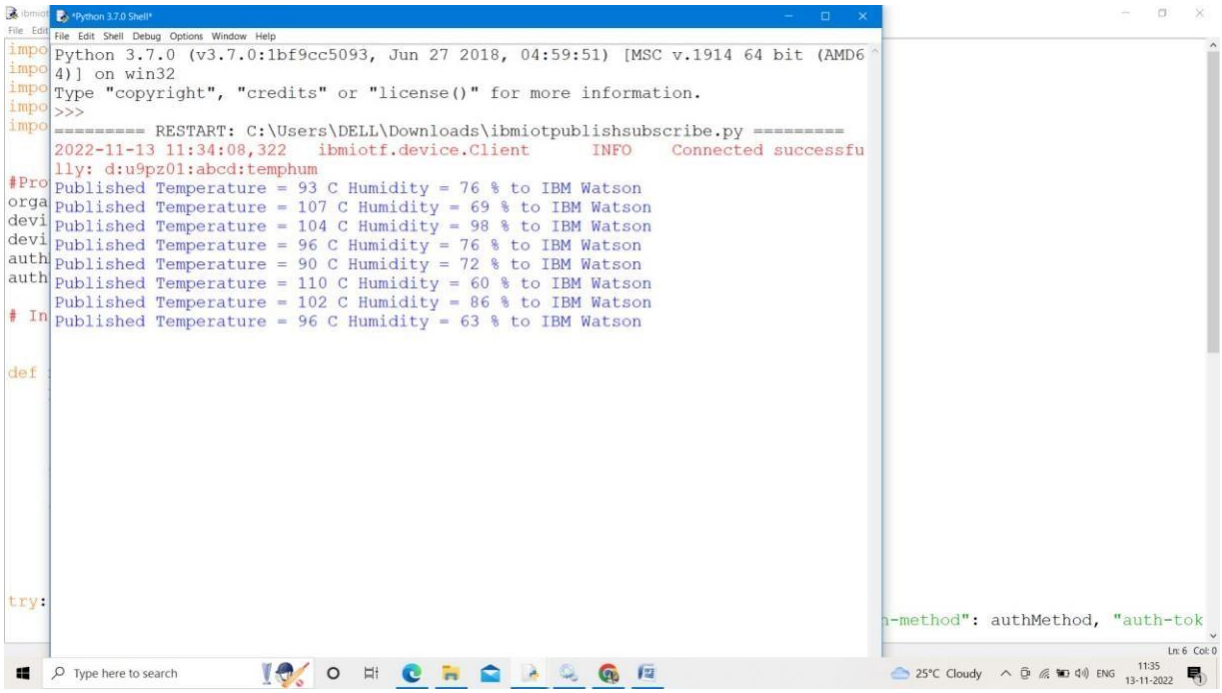
```
        print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "to
IBM Watson")


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

      if not success:

         print("Not connected to IoTF")

time.sleep(10)


      deviceCli.commandCallback = myCommandCallback


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

Publish data to IBM Cloud:




sprint 1.py - C:/Users/dell/Documents/ASSIGNMENTS/sprint 1.py (3.7.4)
File Edit Format Run Options Window Help
Python Shell
Check Module Alt+X
Run Module F5
Run... Customized Shift+F5
import time
import sys
import ibmiotf
import ibmiotf
import random
#Provide your IBM Watson Device Credentials
organization = "wlp5bv"
deviceType = "nsps"
deviceId = "nsp_1"
authMethod = "token"
authToken = "EWGwG15F6EKUtFh5W_"
# 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:
Activate Windows
Go to Settings to activate Windows.
Ln: 1 Col: 11
Type here to search
10:16 PM
11/16/2022

Code Output:


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\DELL\Downloads\ibmiotpublishsubscribe.py ==========
2022-11-13 11:34:08,322 ibmiotf.device.Client INFO Connected successfully: d:u9pz01:abcd:temphum
Published Temperature = 93 C Humidity = 76 % to IBM Watson
Published Temperature = 107 C Humidity = 69 % to IBM Watson
Published Temperature = 104 C Humidity = 98 % to IBM Watson
Published Temperature = 96 C Humidity = 76 % to IBM Watson
Published Temperature = 90 C Humidity = 72 % to IBM Watson
Published Temperature = 110 C Humidity = 60 % to IBM Watson
Published Temperature = 102 C Humidity = 86 % to IBM Watson
Published Temperature = 96 C Humidity = 63 % to IBM Watson
Ln: 6 Col: 0
Type here to search
25°C Cloudy
ENG
11:35
13-11-2022

IBM Watson Output: