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

Team id: PNT2022TMID33893

Project Name: Gas Leakage Monitoring and Alerting System

Develop python code:

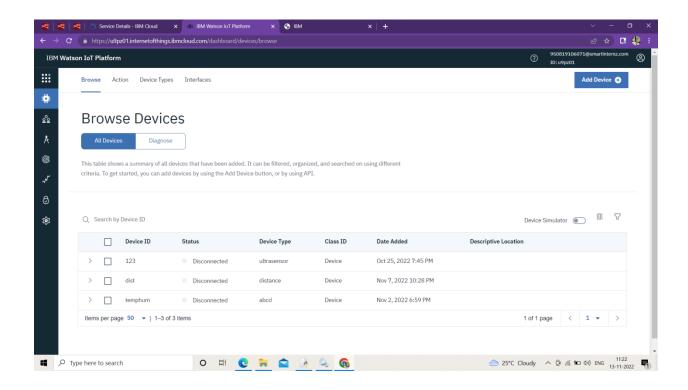
```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
organization = "u9pz01"
deviceType = "abcd"
deviceId = "temphum"
authMethod = "token"
authToken = "12345678"
# Initialize GPIO
def myCommandCallback(cmd):
  print("Command 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:

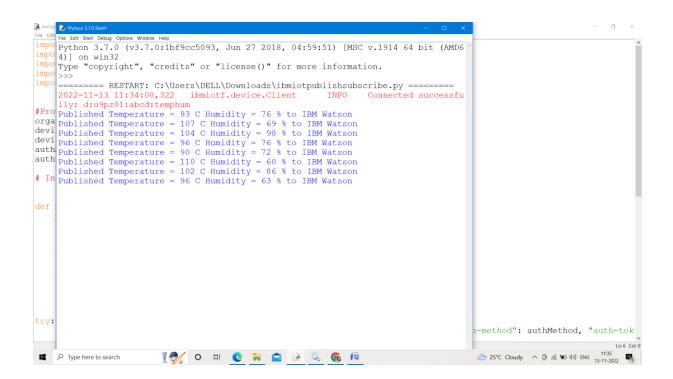
Step 1 : Open IBM WATSON IOT PLATFORM from IBM catalog.



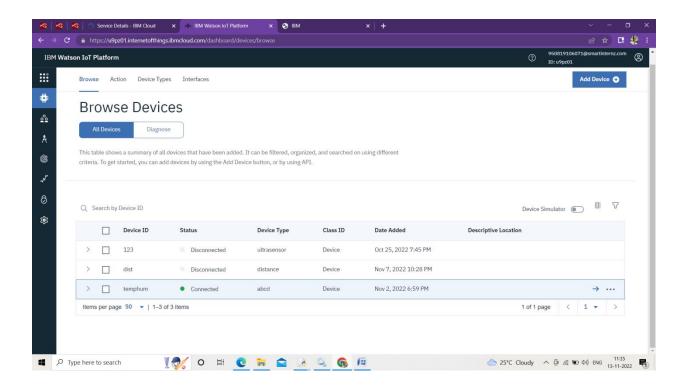
Step 2: Open IDLE Python 3.7.0 and Run the Python code.

```
import t Python Shell Check Module Alt+X
import s Check Module Alt+X Run Module F5 import ibmioti.appircation
 import ibmiotf.device
import random
#Provide your IBM Watson Device Credentials
#Provide your IBM watson
organization = "uppz01"
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)
           deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-tok
deviceCli = ibmiotf device Client(deviceOptions)
                                                                                                                          Type here to search
                                   【 🦭 O 🖽 🥲 🙀 😩 🔞 🙉
```

Step 3: The random values for Temperature and Humidity are produced in the output. And the data is send to the IBM Watson IOT Platform.



Step 4 : In IBM Watson IOT Platform the status shows connected when the python code is made to run.



Step 5 : On clicking Recent Events we can see the Temperature and Humidity values from Python code is published to the IBM Watson IOT Platform.

