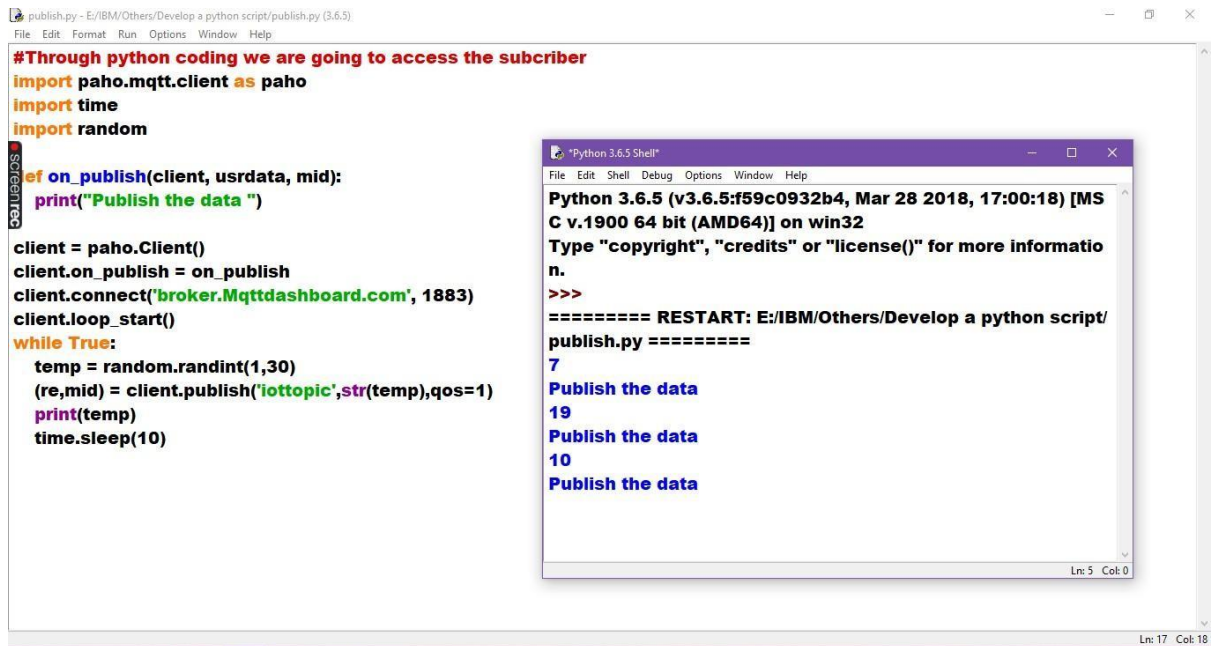


# Publish Data to the IBM Cloud

Date	7 November 2022
Team ID	PNT2022TMID22779
Project Name	Project – Signs with Smart Connectivity for Better Road Safety

## Signs with smart connectivity for Better road safety



The screenshot shows a Python script in a text editor and its execution in a shell. The script, named `publish.py`, is designed to publish data to an MQTT broker. It includes a comment: `#Through python coding we are going to access the subscriber`. The script imports `paho.mqtt.client as paho`, `time`, and `random`. It defines a function `on_publish` that prints "Publish the data ". The main logic creates a `paho.Client()`, connects to `broker.mqttdashboard.com` on port 1883, and enters a `while True` loop. In the loop, it generates a random integer between 1 and 30, publishes it to the topic `iottopic` with a QoS of 1, prints the value, and sleeps for 10 seconds.

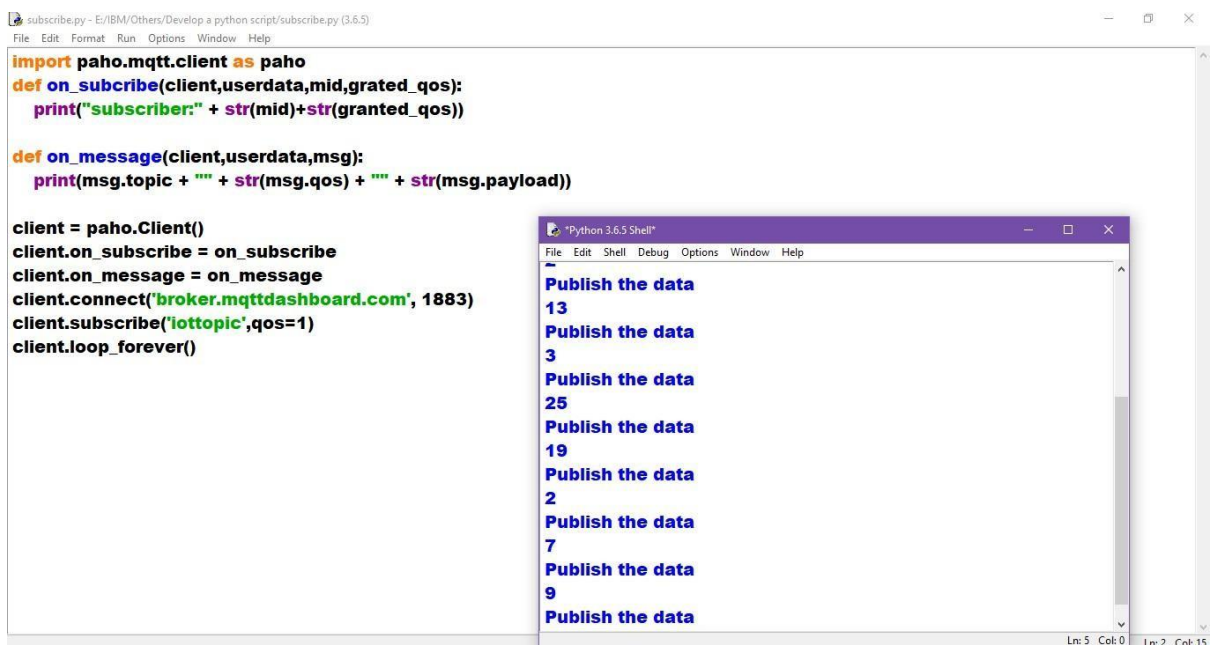
```
#Through python coding we are going to access the subscriber
import paho.mqtt.client as paho
import time
import random

def on_publish(client, userdata, mid):
    print("Publish the data ")

client = paho.Client()
client.on_publish = on_publish
client.connect('broker.mqttdashboard.com', 1883)
client.loop_start()
while True:
    temp = random.randint(1,30)
    (re,mid) = client.publish('iottopic',str(temp),qos=1)
    print(temp)
    time.sleep(10)
```

The shell output shows the execution of the script, displaying the message "Publish the data" followed by the random values 7, 19, 10, and 10.

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MS
C v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more informatio
n.
>>>
===== RESTART: E:/IBM/Others/Develop a python script/
publish.py =====
7
Publish the data
19
Publish the data
10
Publish the data
```



The screenshot shows a Python script in a text editor and its execution in a shell. The script, named `subscribe.py`, is designed to subscribe to an MQTT broker. It imports `paho.mqtt.client as paho`. It defines a function `on_subscribe` that prints "subscriber:" followed by the mid and granted\_qos. It also defines a function `on_message` that prints the topic, qos, and payload. The main logic creates a `paho.Client()`, connects to `broker.mqttdashboard.com` on port 1883, subscribes to the topic `iottopic` with a QoS of 1, and calls `client.loop_forever()`.

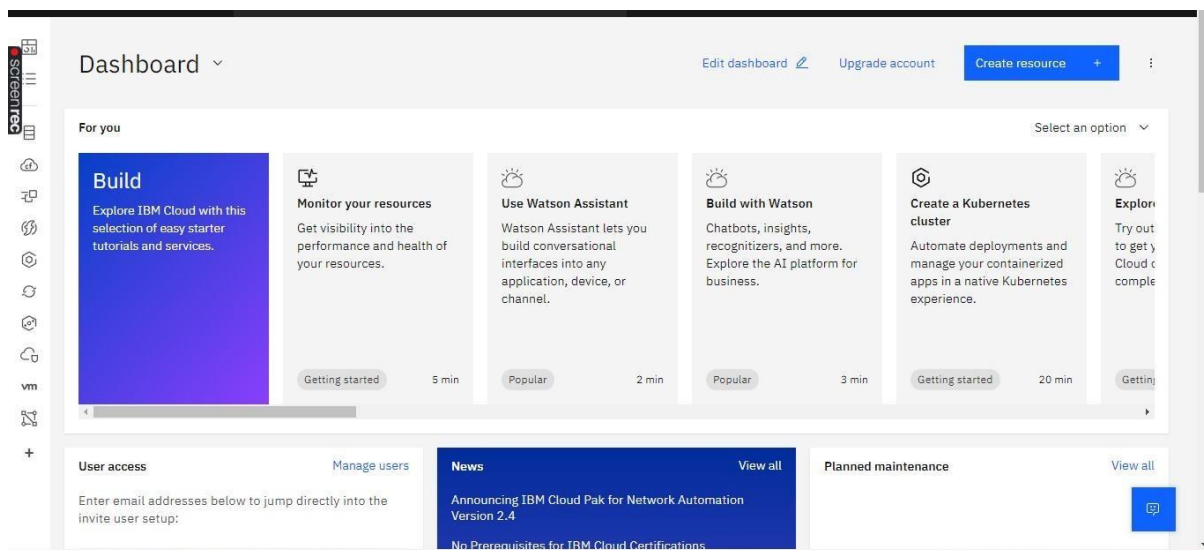
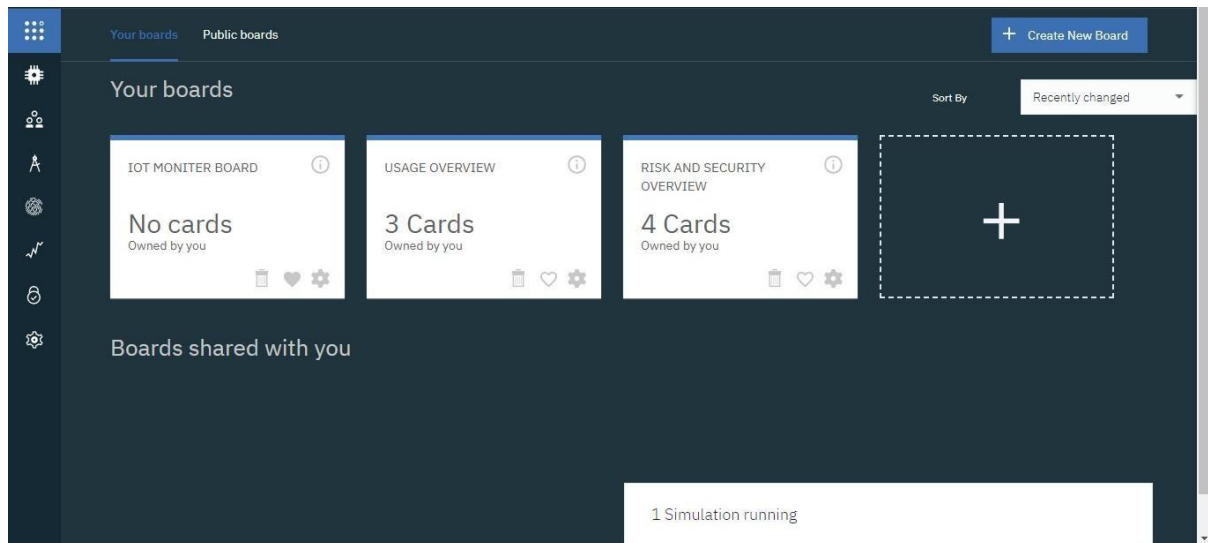
```
import paho.mqtt.client as paho
def on_subscribe(client,userdata,mid,grated_qos):
    print("subscriber:" + str(mid)+str(granted_qos))

def on_message(client,userdata,msg):
    print(msg.topic + "" + str(msg.qos) + "" + str(msg.payload))

client = paho.Client()
client.on_subscribe = on_subscribe
client.on_message = on_message
client.connect('broker.mqttdashboard.com', 1883)
client.subscribe('iottopic',qos=1)
client.loop_forever()
```

The shell output shows the execution of the script, displaying the message "Publish the data" followed by the random values 13, 3, 25, 19, 2, 7, 9, and 9.

```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Publish the data
13
Publish the data
3
Publish the data
25
Publish the data
19
Publish the data
2
Publish the data
7
Publish the data
9
Publish the data
```



## Program :

#IBM Watson IOT Platform

#pip install wiotp-sdk

import wiotp.sdk.device

import time

import random

```

myConfig = {
    "identity": {
        "orgId": "hj5fmy",
        "typeId": "NodeMCU",
        "deviceId": "12345" },
    "auth": { "token": "12345678" }
}
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()

```