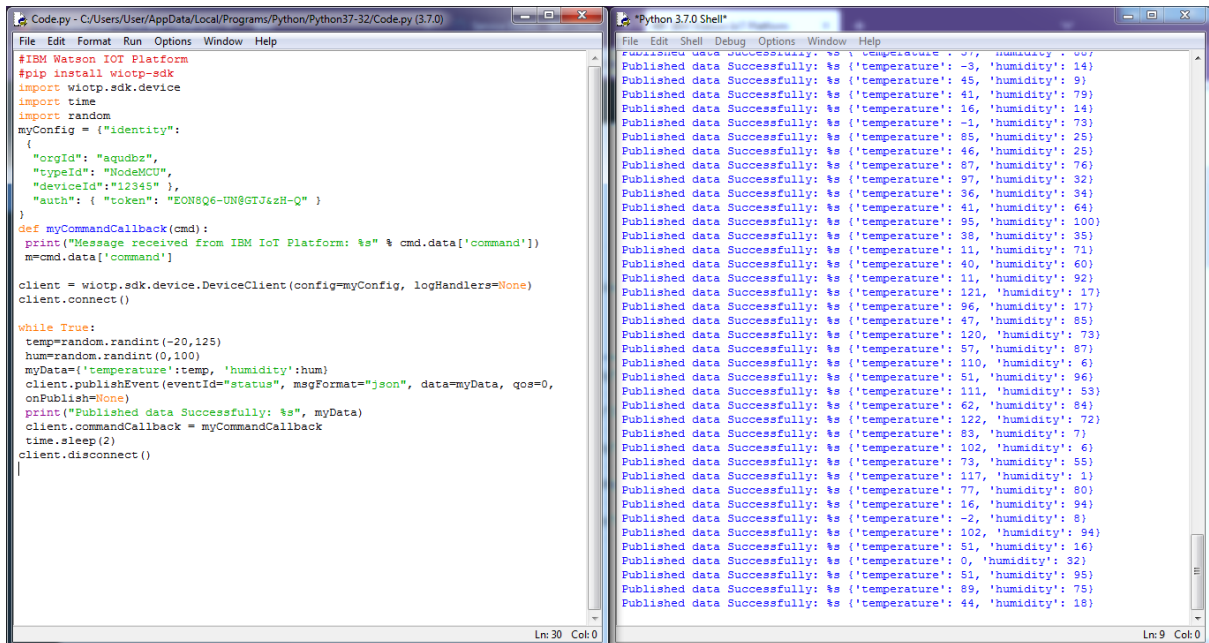


# PUBLISH DATA TO THE IBM CLOUD



The image shows two windows from a Windows operating system. The left window is a code editor titled 'Code.py - C:/Users/User/AppData/Local/Programs/Python/Python37-32/Code.py (3.7.0)'. It contains a Python script that uses the 'wiottp-sdk' to connect to the IBM Watson IoT Platform and publish simulated temperature and humidity data. The script includes a configuration dictionary, a command callback function, and a loop that generates random data and publishes it. The right window is a 'Python 3.7.0 Shell' showing the output of the script, which consists of multiple lines of 'Published data Successfully' messages, each containing a JSON object with 'temperature' and 'humidity' values.

```
#IBM Watson IoT Platform
#pip install wiottp-sdk
import wiottp.sdk.device
import time
import random

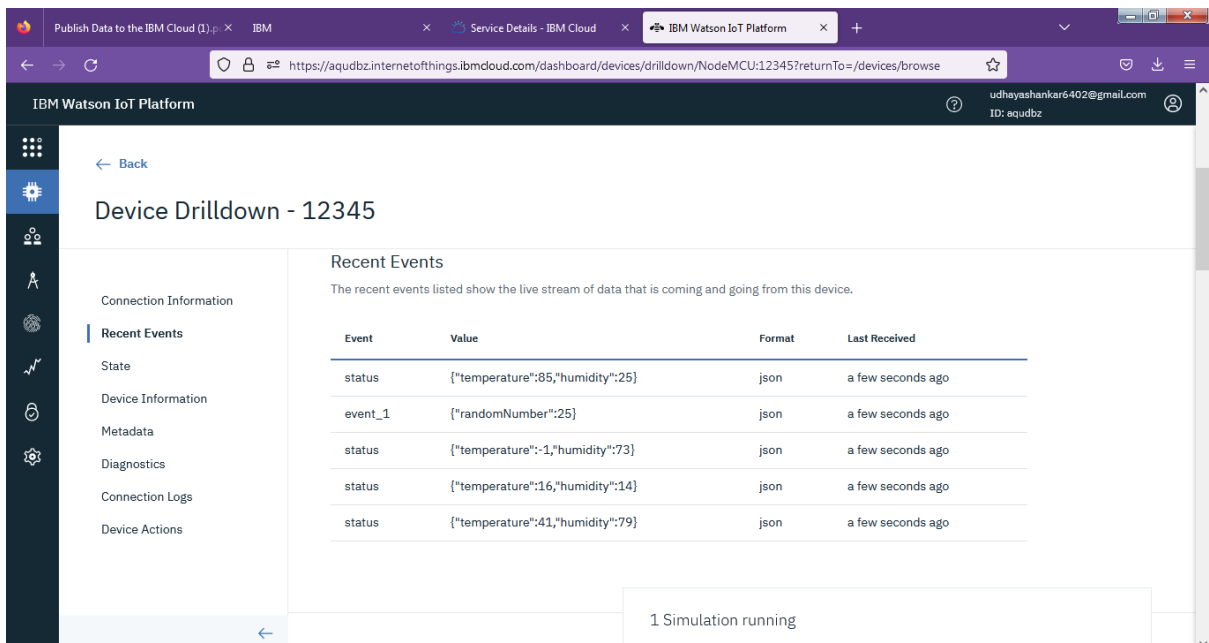
myConfig = {"identity":
{
    "orgId": "aqudbz",
    "typeId": "NodeMCU",
    "deviceId": "12345",
    "auth": { "token": "EON8Q6-UN@GTJ6zH-Q" }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    m=cmd.data['command']

client = wiottp.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()
```

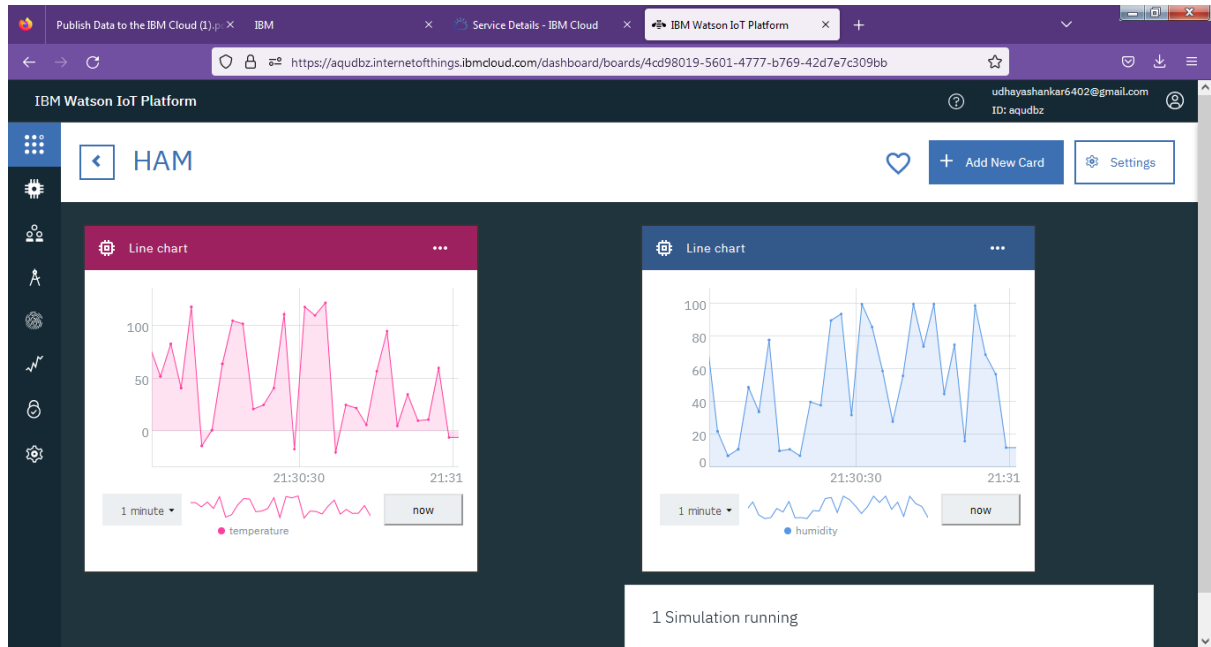
```
Published data Successfully: %s ('temperature': 37, 'humidity': 60)
Published data Successfully: %s ('temperature': -3, 'humidity': 14)
Published data Successfully: %s ('temperature': 45, 'humidity': 9)
Published data Successfully: %s ('temperature': 41, 'humidity': 79)
Published data Successfully: %s ('temperature': 16, 'humidity': 14)
Published data Successfully: %s ('temperature': -1, 'humidity': 73)
Published data Successfully: %s ('temperature': 85, 'humidity': 25)
Published data Successfully: %s ('temperature': 46, 'humidity': 25)
Published data Successfully: %s ('temperature': 87, 'humidity': 76)
Published data Successfully: %s ('temperature': 97, 'humidity': 32)
Published data Successfully: %s ('temperature': 36, 'humidity': 34)
Published data Successfully: %s ('temperature': 41, 'humidity': 64)
Published data Successfully: %s ('temperature': 95, 'humidity': 100)
Published data Successfully: %s ('temperature': 38, 'humidity': 35)
Published data Successfully: %s ('temperature': 11, 'humidity': 71)
Published data Successfully: %s ('temperature': 40, 'humidity': 60)
Published data Successfully: %s ('temperature': 11, 'humidity': 92)
Published data Successfully: %s ('temperature': 121, 'humidity': 17)
Published data Successfully: %s ('temperature': 96, 'humidity': 17)
Published data Successfully: %s ('temperature': 47, 'humidity': 85)
Published data Successfully: %s ('temperature': 120, 'humidity': 73)
Published data Successfully: %s ('temperature': 57, 'humidity': 87)
Published data Successfully: %s ('temperature': 110, 'humidity': 6)
Published data Successfully: %s ('temperature': 51, 'humidity': 96)
Published data Successfully: %s ('temperature': 111, 'humidity': 53)
Published data Successfully: %s ('temperature': 62, 'humidity': 84)
Published data Successfully: %s ('temperature': 122, 'humidity': 72)
Published data Successfully: %s ('temperature': 83, 'humidity': 7)
Published data Successfully: %s ('temperature': 102, 'humidity': 6)
Published data Successfully: %s ('temperature': 73, 'humidity': 55)
Published data Successfully: %s ('temperature': 117, 'humidity': 1)
Published data Successfully: %s ('temperature': 77, 'humidity': 80)
Published data Successfully: %s ('temperature': 16, 'humidity': 94)
Published data Successfully: %s ('temperature': -2, 'humidity': 8)
Published data Successfully: %s ('temperature': 102, 'humidity': 94)
Published data Successfully: %s ('temperature': 51, 'humidity': 16)
Published data Successfully: %s ('temperature': 0, 'humidity': 32)
Published data Successfully: %s ('temperature': 51, 'humidity': 95)
Published data Successfully: %s ('temperature': 89, 'humidity': 75)
Published data Successfully: %s ('temperature': 44, 'humidity': 18)
```



The image shows a web browser window displaying the IBM Watson IoT Platform dashboard. The browser tabs include 'Publish Data to the IBM Cloud (1)', 'IBM', 'Service Details - IBM Cloud', and 'IBM Watson IoT Platform'. The address bar shows the URL 'https://aqudbz.internetofthings.ibmcloud.com/dashboard/devices/drilldown/NodeMCU:12345?returnTo=/devices/browse'. The dashboard header shows the user 'udhayashankar6402@gmail.com' with ID 'aqudbz'. The main content area is titled 'Device Drilldown - 12345' and features a sidebar with navigation options: 'Connection Information', 'Recent Events', 'State', 'Device Information', 'Metadata', 'Diagnostics', 'Connection Logs', and 'Device Actions'. The 'Recent Events' section is active, displaying a table of recent events. Below the table, a status box indicates '1 Simulation running'.

Event	Value	Format	Last Received
status	{"temperature":85,"humidity":25}	json	a few seconds ago
event_1	{"randomNumber":25}	json	a few seconds ago
status	{"temperature":-1,"humidity":73}	json	a few seconds ago
status	{"temperature":16,"humidity":14}	json	a few seconds ago
status	{"temperature":41,"humidity":79}	json	a few seconds ago

1 Simulation running



## PROGRAM:

#IBM Watson IOT Platform

#pip install wiotp-sdk

```
import wiotp.sdk.device
```

```
import time
```

```
import random
```

```
myConfig = {"identity":
```

```
{
```

```
    "orgId": "aqudbz",
```

```
    "typeId": "NodeMCU",
```

```
    "deviceId": "12345" },
```

```
"auth": { "token": "EON8Q6-UN@GTJ&zH-Q" }  
}  
  
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()
```