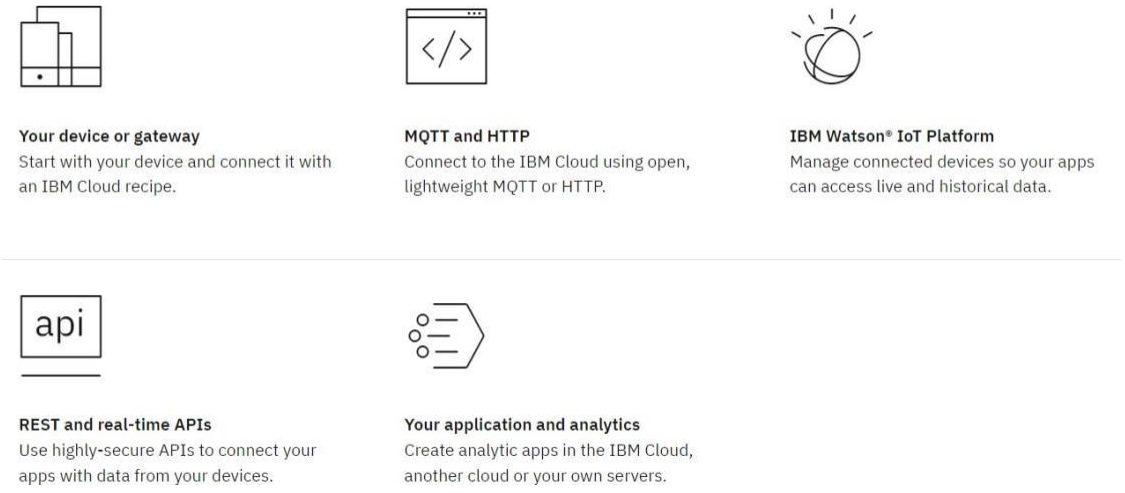
Sprint- 2

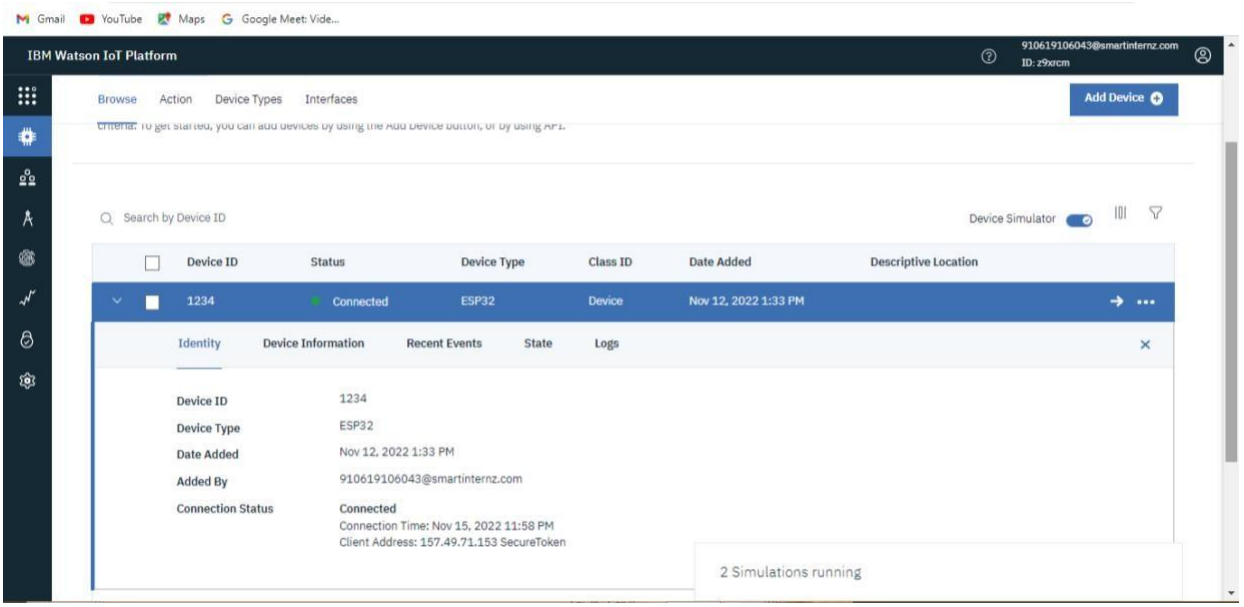
Team ID	**PNT2022TMID17768**
Project Title	**Gas Leakage Monitoring And Alerting System**
Date	**15.11.2022**

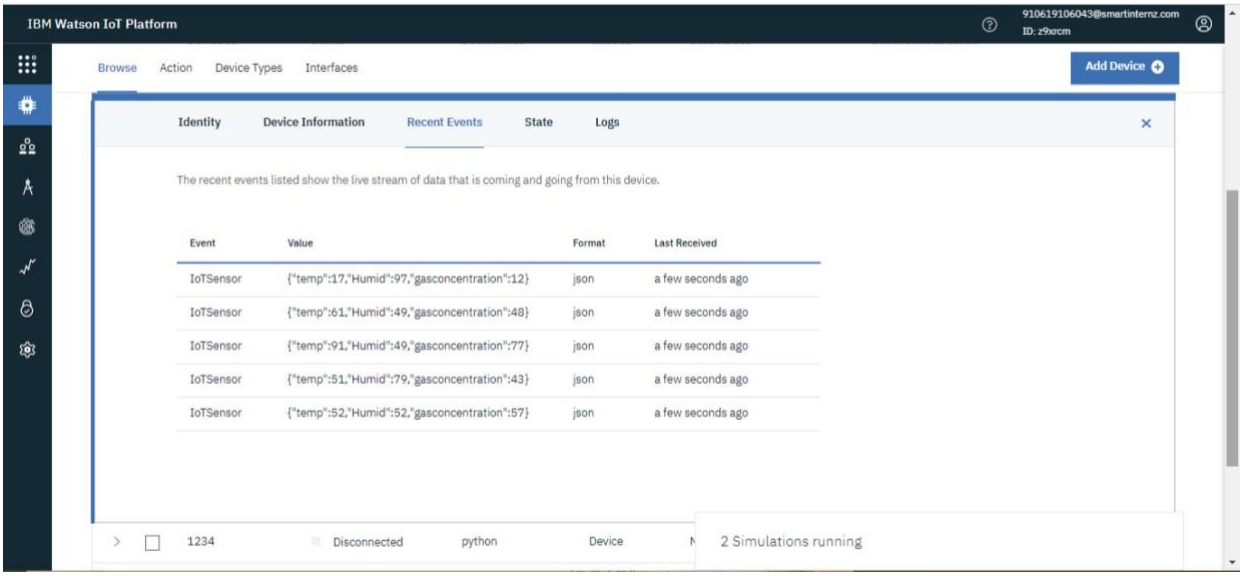
IBM Watson and Python Integration:

By using Watson IoT Platform, you can collect connected device data and perform analytics on real-time data. The IBM Watson IoT Platform is a fully managed, Cloud-hosted service that provides device management capabilities as well as datacollection and management in a time series format.



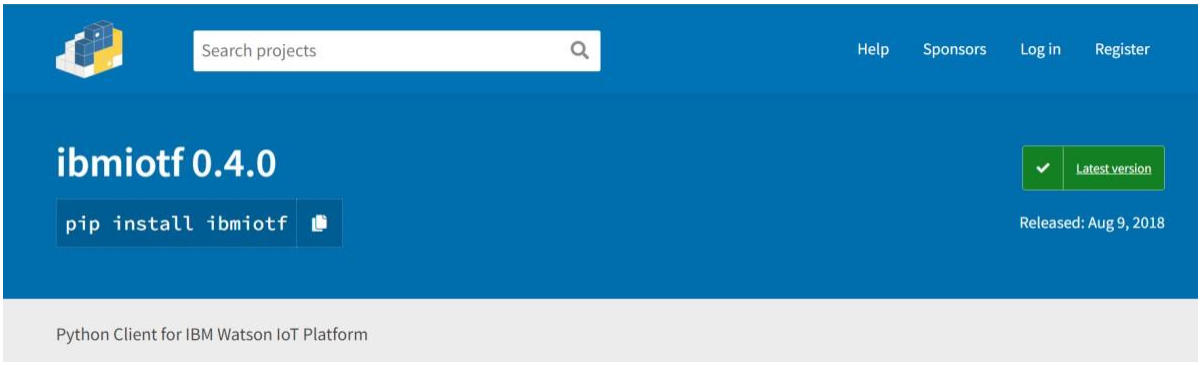
Using the Device Created in IBM Watson:





Connected sign shows that it is connected and live

Python code execution:



Install this package : Python Client for IBM Watson IoT Platform

python code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random



#Provide your IBM Watson Device Credentials
organization = "z9xrcm"
deviceType = "ESP32"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="sprinkleron":
        print ("Sprinkler is on")
    else :
        print ("Sprinkler is off")

    #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 eventof type "greeting" 10 times
deviceCli.connect()

while True:
        #Get Sensor Data from DHT11

        temp=random.randint(0,100)
        Humid=random.randint(0,100)
        gasconcentration=random.randint(0,100)



        data = { 'temp' : temp, 'Humid': Humid, "gasconcentration": gasconcentration}#print

        data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "gasconcentration = %s %%" % gasconcentration, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback

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

code.py - C:\Users\bala\AppData\Local\Programs\Python\Python36-32\code.py (3.6.0)

File Edit Format Run Options Window Help

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random


#Provide your IBM Watson Device Credentials
organization = "z9xrcm"
deviceType = "ESP32"
deviceId = "1234"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="sprinkleron":
        print ("Sprinkler is on")
    else :
        print ("Sprinkler is off")

    #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
```

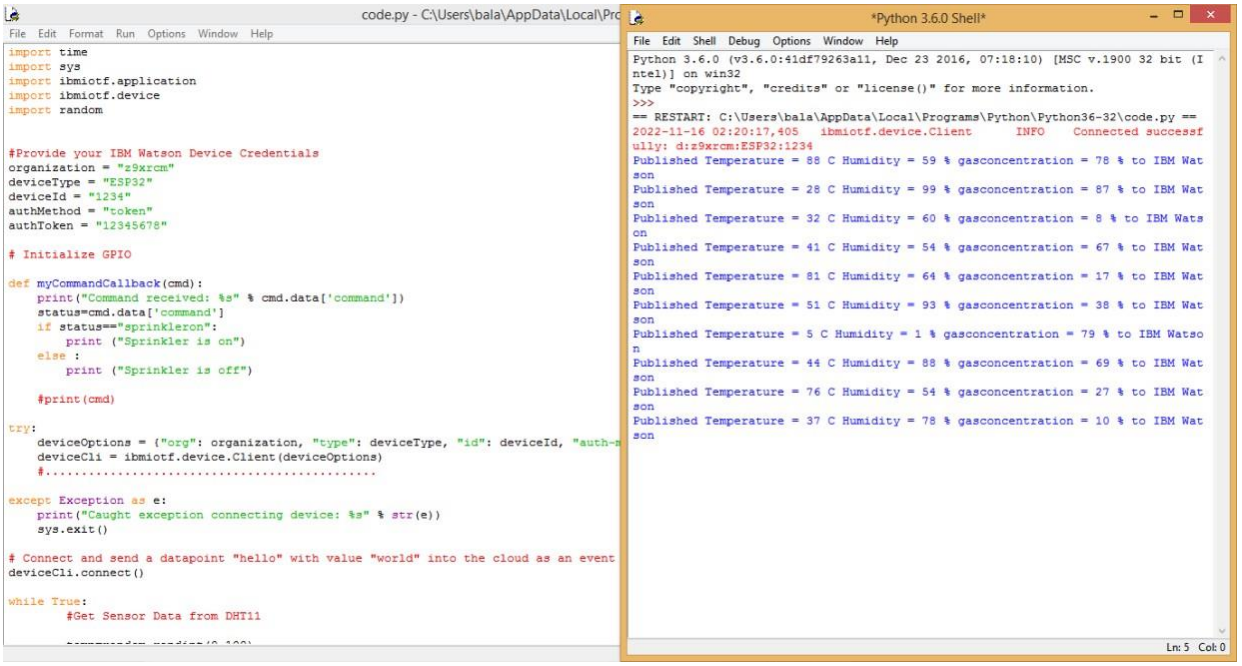
Ln: 57 Col: 0

code.py - C:\Users\bala\AppData\Local\Programs\Python\Python36-32\code.py (3.6.0)

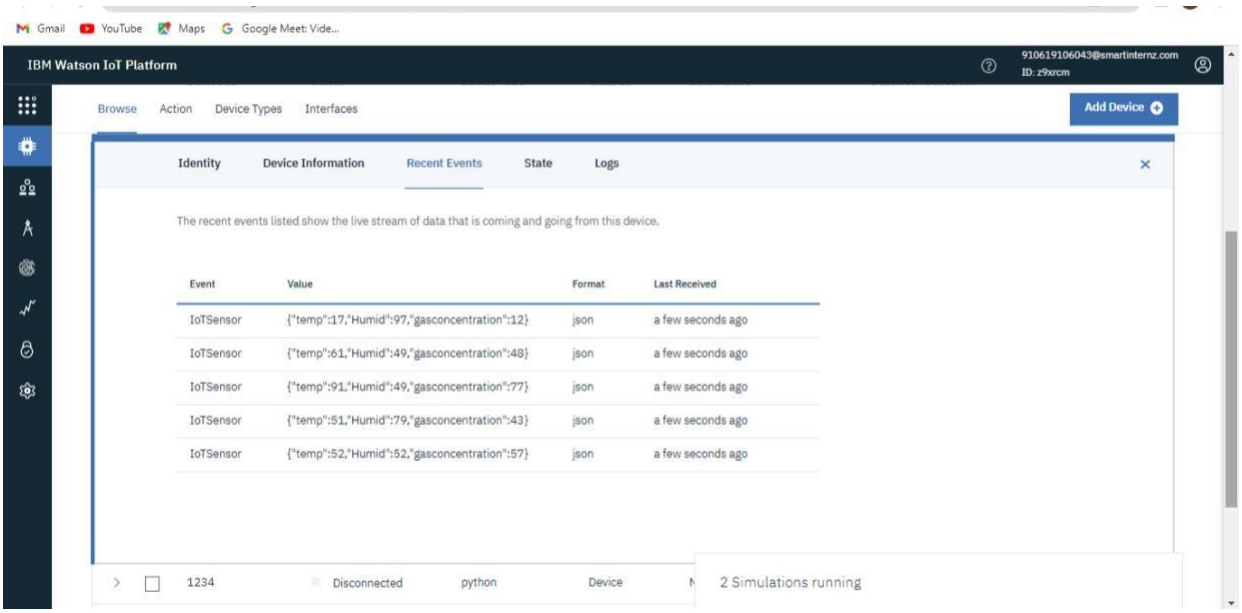
File Edit Format Run Options Window Help

```
    #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(0,100)
        Humid=random.randint(0,100)
        gasconcentration=random.randint(0,100)


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

        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "gasconcentration = %s %%" % gasconcentration, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(1)

        deviceCli.commandCallback = myCommandCallback

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

Ln: 57 Col: 0



Recent Events in IBM Watson IoT Platform:



Boards in IBM Platform: