

PROJECT DEVELOPMENT PHASE

SPRINT 2

Date	18 November 2022
Team ID	PNT2022TMID13514
Project name	Gas Leakage Monitoring & Alerting System for Industries

WATSON REFERENCE LINK: <https://l27fmg.internetofthings.ibmcloud.com/dashboard/boards/406e3189-58ac-447d-b546-59ec0b8a582d>

IBM WATSON :

IBM Watson IoT Platform

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
123456	Connected	12345	Device	Nov 12, 2022 10:49 AM	

Items per page 50 | 1-1 of 1 item

2 Simulations running

IBM Watson IoT Platform

Your boards

Create New Board

Sort By Recently changed

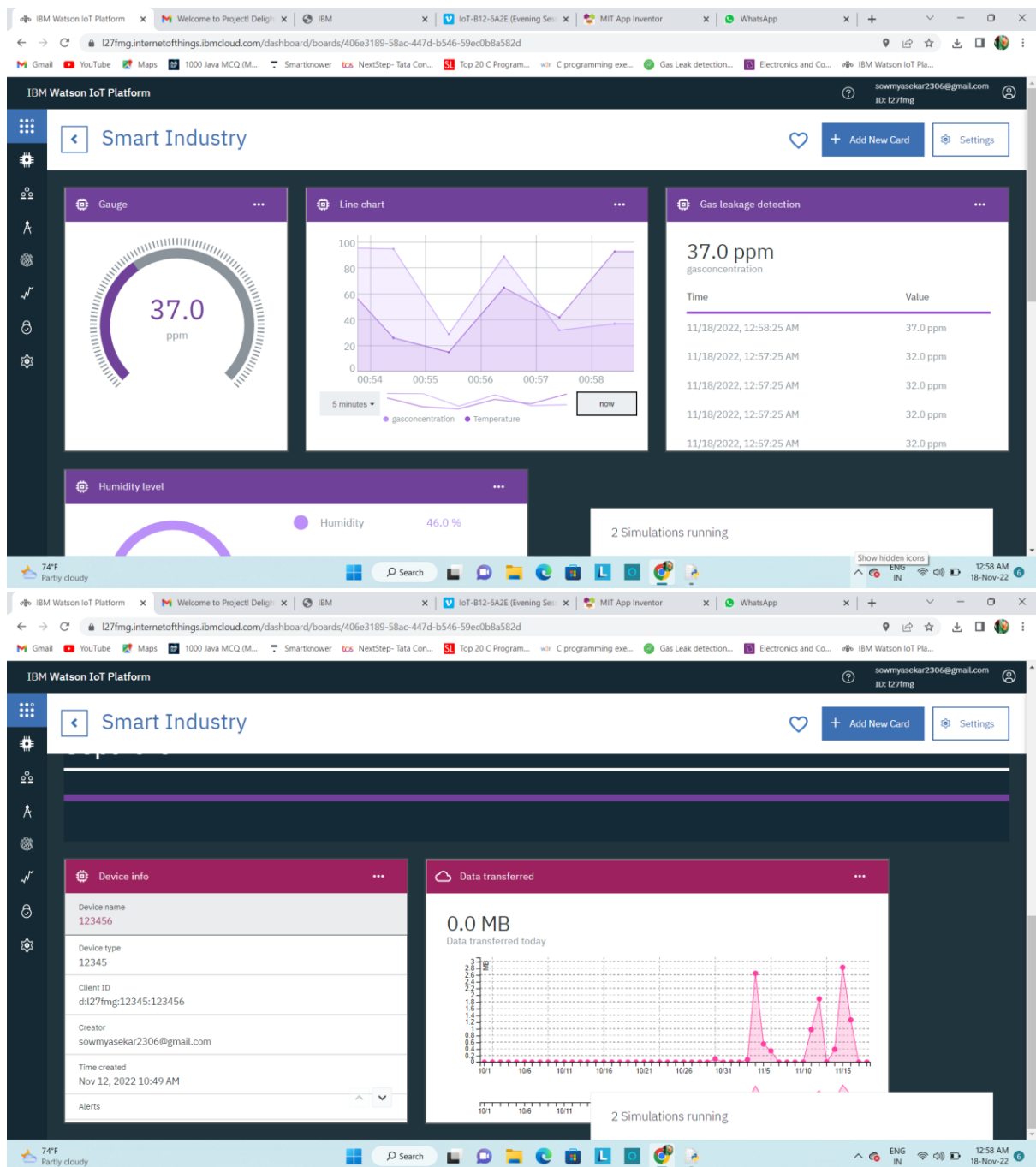
SMART INDUSTRY 7 Cards Owned by you

RISK AND SECURITY OVERVIEW 4 Cards Owned by you

USAGE OVERVIEW 3 Cards Owned by you

Boards shared with you

2 Simulations running



PYTHON CODE CONNECTED WITH WATSON :

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

organization = "l27fmg"

```

deviceType = "12345"

deviceId = "123456"

authMethod ="token"

authToken = "123456789"

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")

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()

deviceCli.connect()

while True:

    gasconcentration = random.randint(90,110)

    Humidity =random.randint(90,110)

    Temperature = random.randint(90,110)

    data = {'gasconcentration' : gasconcentration, 'Humidity' : Humidity, 'Temperature'
:Temperature}

    def myOnPublishCallback():

```

print(" GasConcentration = %s PPM" % gasconcentration, "to IBM Watson")

print(" Humidity = %s%%" % Humidity, "to IBM Watson")

print(" Temperature = %s C" % Temperature, "to IBM Watson")

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

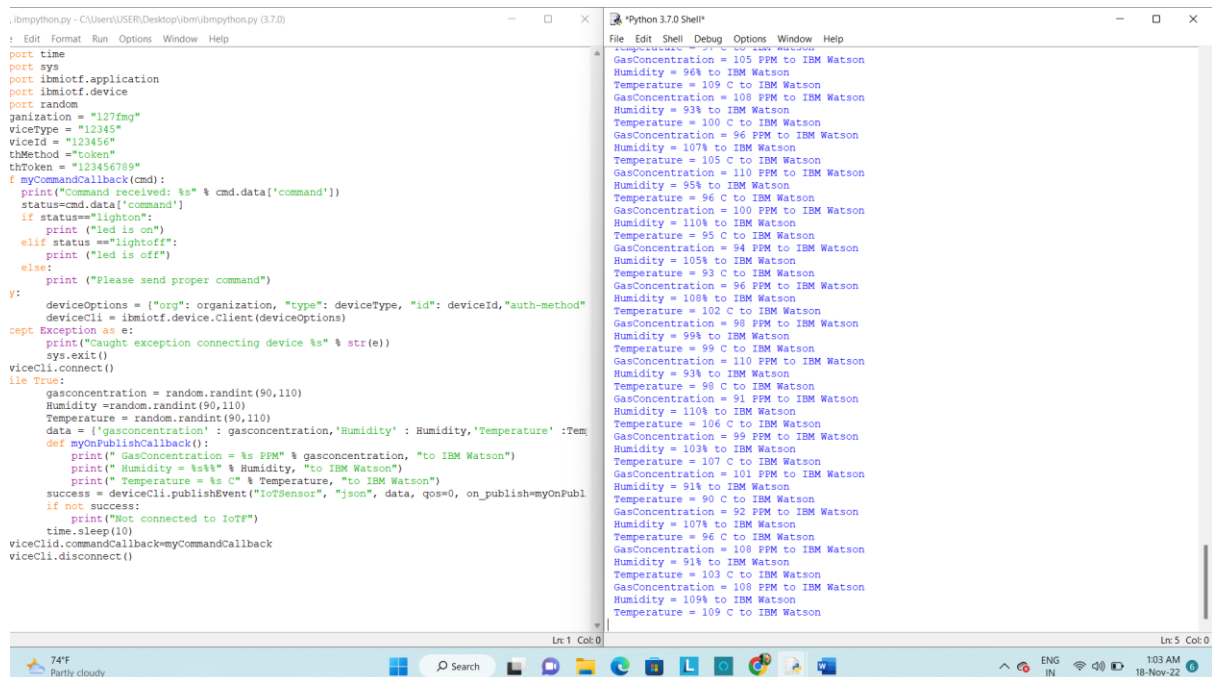
if not success:

print("Not connected to IoTF")

time.sleep(10)

deviceCliId.commandCallback=myCommandCallback

deviceCli.disconnect()



```
.\bmppython.py - C:\Users\USER\Desktop\ibm\bmppython.py (3.7.0)
: Edit Format Run Options Window Help
port time
port sys
port ibmiotf.application
port ibmiotf.device
port random
janization = "127fmg"
viceType = "12345"
viceId = "123456"
thMethod = "token"
thToken = "123456789"
f 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")
Y:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method"
deviceCli = ibmiotf.device.Client(deviceOptions)
cept Exception as e:
    print("Caught exception connecting device %s" % str(e))
    sys.exit()
viceCli.connect()
ile True:
    gasconcentration = random.randint(90,110)
    Humidity = random.randint(90,110)
    Temperature = random.randint(90,110)
    data = {'gasconcentration': gasconcentration, 'Humidity': Humidity, 'Temperature': Tem
    def myOnPublishCallback():
        print(" GasConcentration = %s PPM" % gasconcentration, "to IBM Watson")
        print(" Humidity = %s%%" % Humidity, "to IBM Watson")
        print(" Temperature = %s C" % Temperature, "to IBM Watson")
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPubl
    if not success:
        print("Not connected to IoTf")
        time.sleep(10)
    viceCliId.commandCallback=myCommandCallback
    viceCli.disconnect()

Ln: 1 Col: 0
```

Python 3.7.0 Shell

```
File Edit Shell Debug Options Window Help
reconnect = 777 to IBM Watson
GasConcentration = 105 PPM to IBM Watson
Humidity = 96% to IBM Watson
Temperature = 109 C to IBM Watson
GasConcentration = 108 PPM to IBM Watson
Humidity = 93% to IBM Watson
Temperature = 100 C to IBM Watson
GasConcentration = 96 PPM to IBM Watson
Humidity = 107% to IBM Watson
Temperature = 105 C to IBM Watson
GasConcentration = 110 PPM to IBM Watson
Humidity = 95% to IBM Watson
Temperature = 96 C to IBM Watson
GasConcentration = 100 PPM to IBM Watson
Humidity = 110% to IBM Watson
Temperature = 95 C to IBM Watson
GasConcentration = 94 PPM to IBM Watson
Humidity = 105% to IBM Watson
Temperature = 93 C to IBM Watson
GasConcentration = 96 PPM to IBM Watson
Humidity = 108% to IBM Watson
Temperature = 102 C to IBM Watson
GasConcentration = 98 PPM to IBM Watson
Humidity = 99% to IBM Watson
Temperature = 99 C to IBM Watson
GasConcentration = 110 PPM to IBM Watson
Humidity = 93% to IBM Watson
Temperature = 98 C to IBM Watson
GasConcentration = 91 PPM to IBM Watson
Humidity = 110% to IBM Watson
Temperature = 106 C to IBM Watson
GasConcentration = 99 PPM to IBM Watson
Humidity = 103% to IBM Watson
Temperature = 107 C to IBM Watson
GasConcentration = 101 PPM to IBM Watson
Humidity = 91% to IBM Watson
Temperature = 90 C to IBM Watson
GasConcentration = 92 PPM to IBM Watson
Humidity = 107% to IBM Watson
Temperature = 96 C to IBM Watson
GasConcentration = 108 PPM to IBM Watson
Humidity = 91% to IBM Watson
Temperature = 103 C to IBM Watson
GasConcentration = 108 PPM to IBM Watson
Humidity = 109% to IBM Watson
Temperature = 109 C to IBM Watson
```

Ln: 5 Col: 0

74°F Partly cloudy

103 AM 18-Nov-22