

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

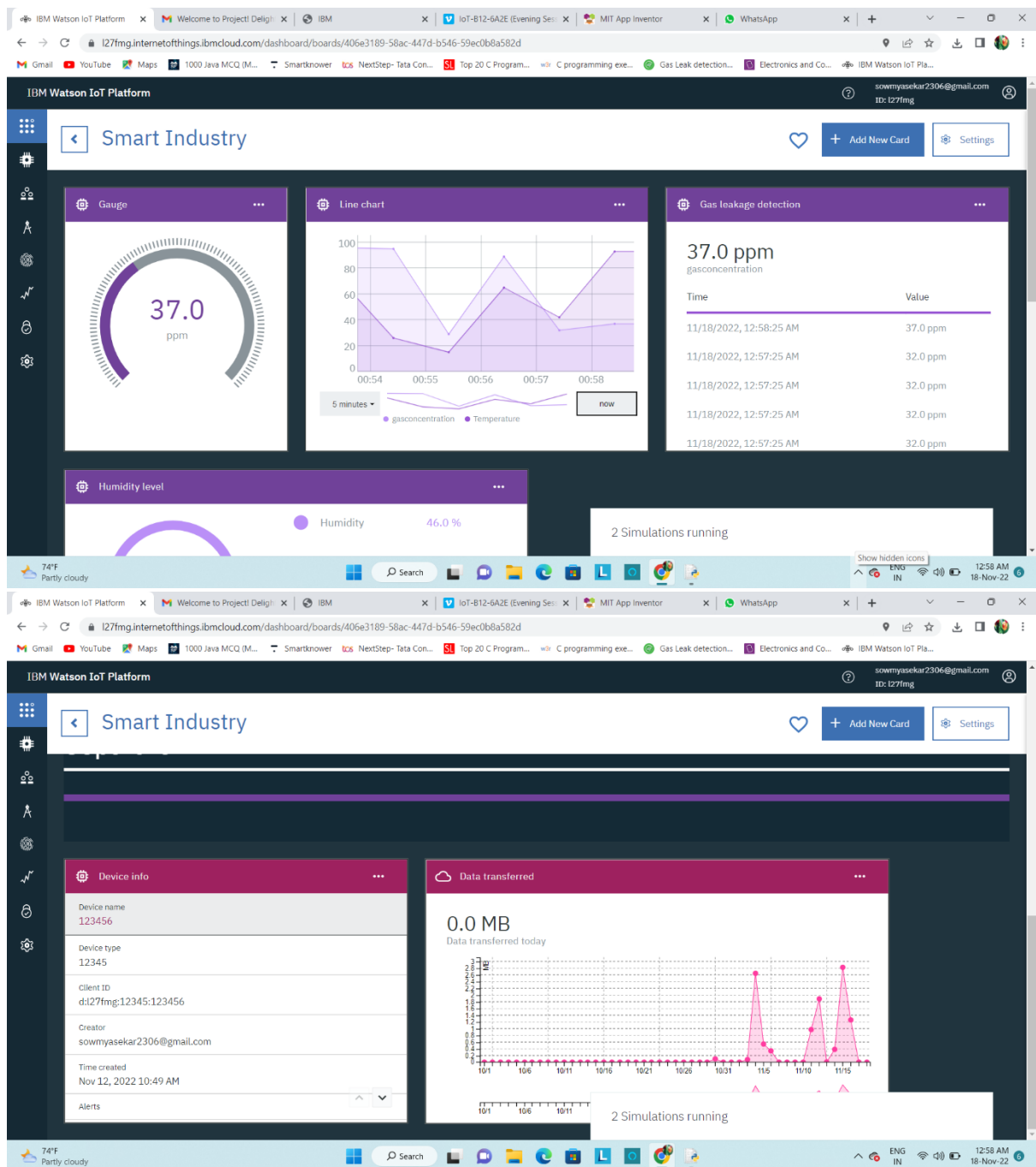
SMART INDUSTRY 7 Cards Owned by you

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 = "I27fmg"

```

deviceType = "12345"

deviceId = "123456"

authMethod ="token"

authToken = "123456789"

def myCommandCallback(cmd):

    print("Command received: %s" % cmd.data['command'])

    status=cmd.data['command']

    if status=="alarmon":

        print ("Alarm is on")

    elif status == "alarmoff":

        print ("Alarm 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(50,110)

    Humidity =random.randint(90,110)

    Temperature = random.randint(90,110)

    if gasconcentration>80:

        gas_status=" Hurry gas is leaking \n Alert!!!!!"

```

```

else:

    gas_status="gas is not leaking"

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

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")

    print(gas_status)

    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

deviceCli.disconnect()

```

The screenshot shows a Python script running in a shell window. The script is titled 'ibmpython.py' and is located at 'C:\Users\USER\Desktop\ibm\ibmpython.py (3.7.0)'. The script is designed to simulate an IoT sensor connected to IBM Watson IoT. It uses the 'ibmiotf' library to connect to the Watson IoT platform. The script defines a 'myCommandCallback' function that handles incoming commands and a 'myOnPublishCallback' function that publishes sensor data to Watson IoT. The sensor data includes 'gasconcentration', 'Humidity', 'Temperature', and 'gas_status'. The script runs a loop that generates random sensor data and publishes it to Watson IoT every 10 seconds. The output of the script is shown in the shell window, displaying the sensor data being published to Watson IoT.

```

ibmpython.py - C:\Users\USER\Desktop\ibm\ibmpython.py (3.7.0)
File Edit Format Run Options Window Help
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
organization = "127fmg"
deviceType = "12345"
deviceId = "123456"
authMethod = "token"
authToken = "123456789"
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status=="alarmon":
        print ("Alarm is on")
    elif status=="alarmoff":
        print ("Alarm is off")
    else:
        print ("Please send proper command")
try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId,"auth-me
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(50,110)
    Humidity =random.randint(90,110)
    Temperature = random.randint(90,110)
    if gasconcentration>80:
        gas_status=" Hurry gas is leaking \n Alert!!!!!"
    else:
        gas_status="gas is not leaking"
    data = {'gasconcentration': gasconcentration,'Humidity': Humidity,'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")
    print(gas_status)
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=my
    if not success:
        print("Not connected to IoTF")
    time.sleep(10)
deviceCli.commandCallback=myCommandCallback

Python Shell
File Edit Shell Debug Options Window Help
Temperature = 92 C to IBM Watson
gas is not leaking
GasConcentration = 72 PPM to IBM Watson
Humidity = 92% to IBM Watson
Temperature = 100 C to IBM Watson
gas is not leaking
GasConcentration = 65 PPM to IBM Watson
Humidity = 103% to IBM Watson
Temperature = 103 C to IBM Watson
gas is not leaking
GasConcentration = 58 PPM to IBM Watson
Humidity = 97% to IBM Watson
Temperature = 99 C to IBM Watson
gas is not leaking
GasConcentration = 71 PPM to IBM Watson
Humidity = 100% to IBM Watson
Temperature = 103 C to IBM Watson
gas is not leaking
GasConcentration = 107 PPM to IBM Watson
Humidity = 103% to IBM Watson
Temperature = 91 C to IBM Watson
Hurry gas is leaking
Alert!!!!
GasConcentration = 98 PPM to IBM Watson
Humidity = 106% to IBM Watson
Temperature = 101 C to IBM Watson
Hurry gas is leaking
Alert!!!!
GasConcentration = 55 PPM to IBM Watson
Humidity = 95% to IBM Watson
Temperature = 95 C to IBM Watson
gas is not leaking
GasConcentration = 87 PPM to IBM Watson
Humidity = 108% to IBM Watson
Temperature = 104 C to IBM Watson
Hurry gas is leaking
Alert!!!!
GasConcentration = 66 PPM to IBM Watson
Humidity = 102% to IBM Watson
Temperature = 110 C to IBM Watson
gas is not leaking
GasConcentration = 78 PPM to IBM Watson
Humidity = 104% to IBM Watson
Temperature = 104 C to IBM Watson
gas is not leaking

```