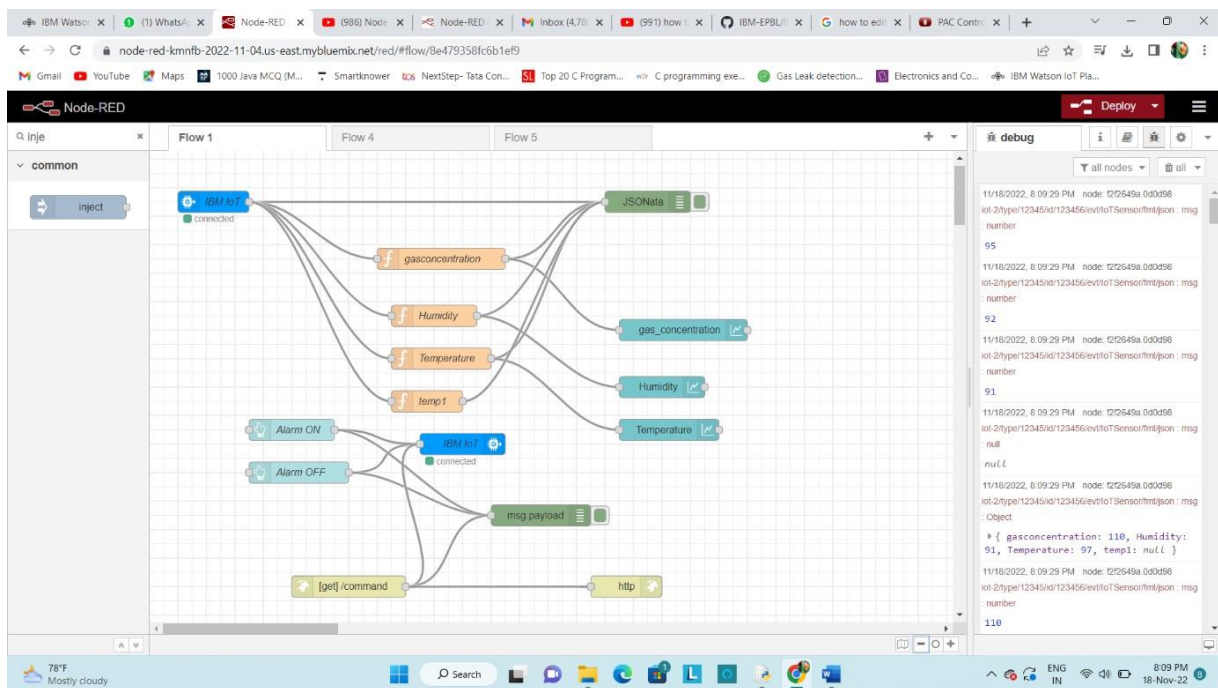


PROJECT DEVELOPMENT PHASE

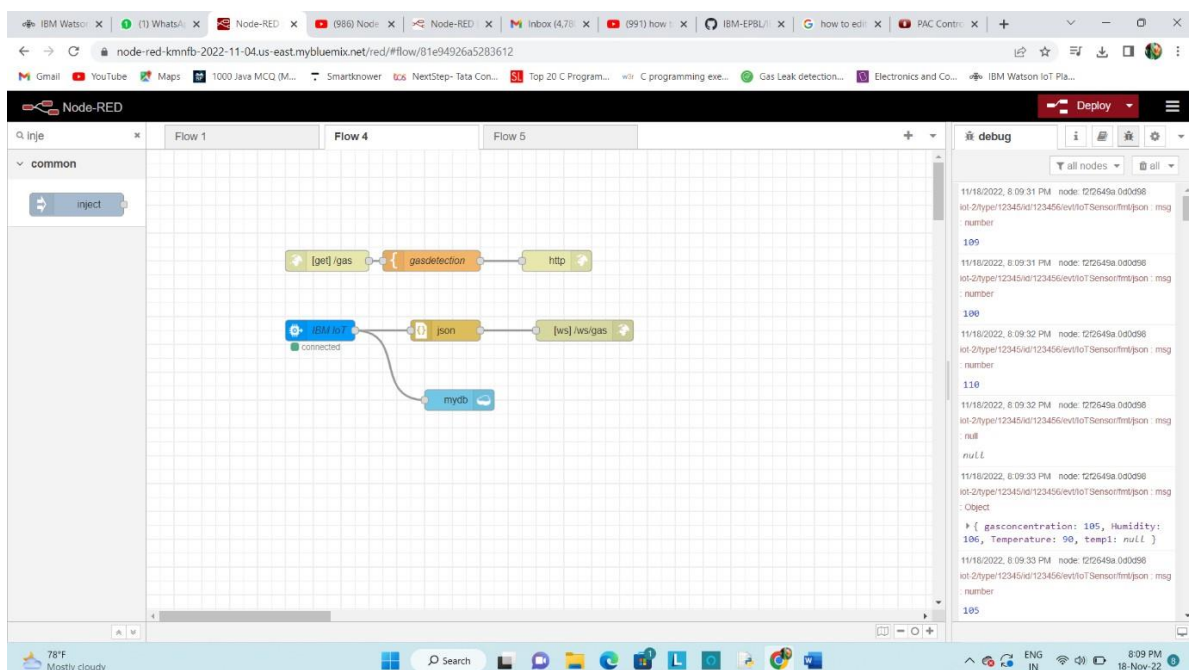
SPRINT 3

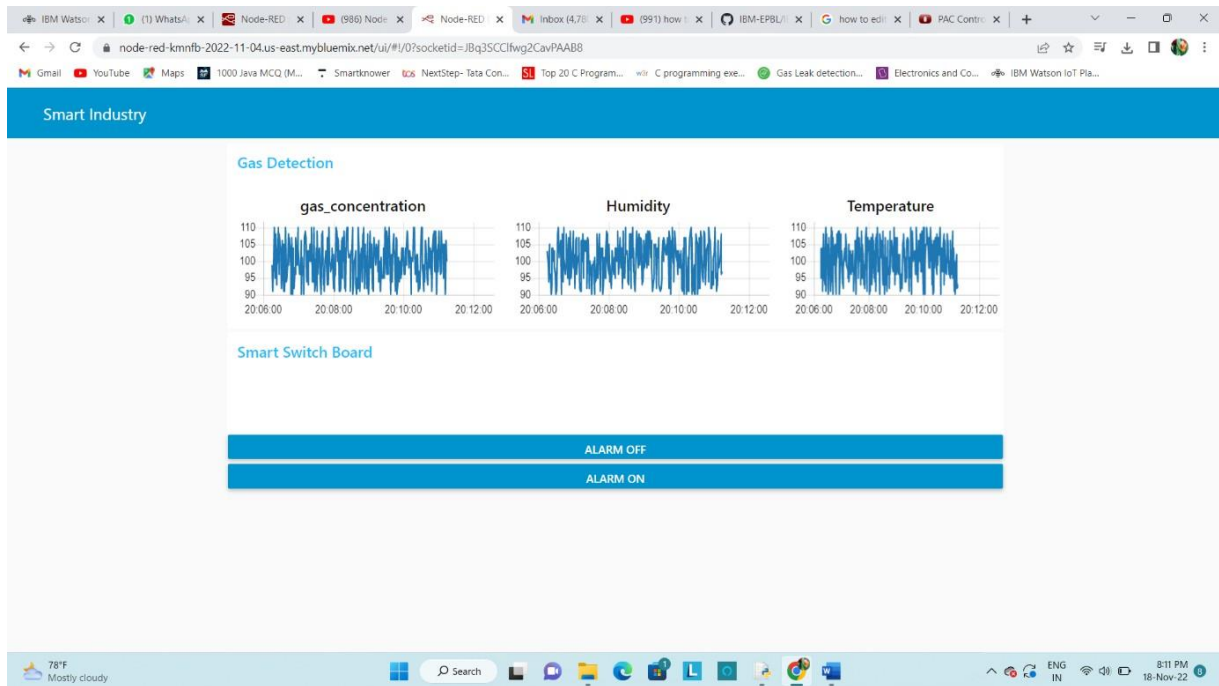
Date	18 November 2022
Team ID	PNT2022TMID13514
Project name	Gas Leakage Monitoring & Alerting System for Industries

NODE RED FLOW



CLOUDANT CONNECTION IN NODE-RED





RECEIVE OF MESSAGE FROM WATSON:

The screenshot shows the IBM Watson IoT Platform dashboard. The top navigation bar includes "Browse", "Action", "Device Types", and "Interfaces". A sidebar on the left contains icons for various platform features. The main content area displays a table of devices. One device, with ID 123456 and status "Connected", is selected. Below the table, the "Recent Events" tab is active, showing a list of events received from the device. The events are displayed in a table with columns for Event, Value, Format, and Last Received.

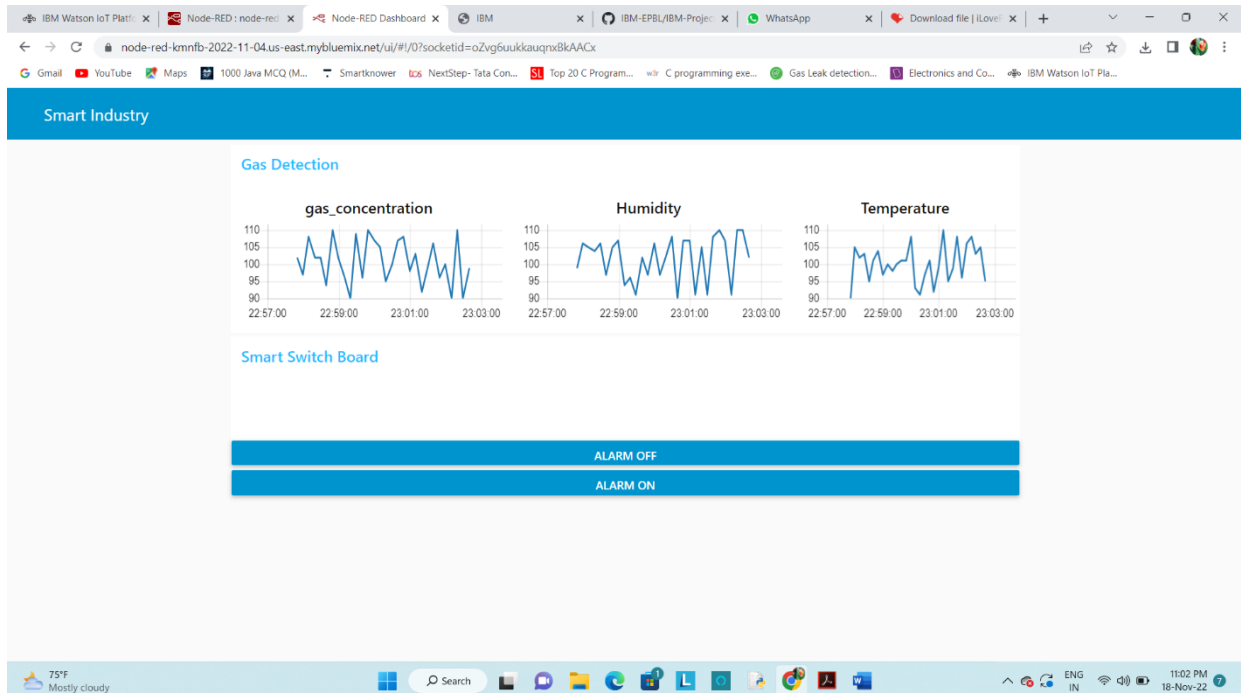
Event	Value	Format	Last Received
IoTSensor	{"gasconcentration":94,"humidity":103,"temperat...	json	a few seconds ago
IoTSensor	{"gasconcentration":108,"humidity":103,"temperat...	json	a few seconds ago
IoTSensor	{"gasconcentration":99,"humidity":100,"temperat...	json	a few seconds ago
IoTSensor	{"gasconcentration":95,"humidity":98,"temperat...	json	a few seconds ago
IoTSensor	{"gasconcentration":99,"humidity":106,"temperat...	json	a few seconds ago

PYTHON CODE OUTPUT:

```
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=="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}
    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 IoT")
    time.sleep(10)
deviceCli.disconnect()

Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
===== RESTART: C:\Users\USER\Desktop\ibm\ibmpython.py =====
2022-11-16 22:09:57,891 ibmiotf.device.Client INFO Connected successfully: d:127fmg:12345:123456
GasConcentration = 110 PPM to IBM Watson
Humidity = 100% to IBM Watson
Temperature = 105 C to IBM Watson
GasConcentration = 98 PPM to IBM Watson
Humidity = 105% to IBM Watson
Temperature = 93 C to IBM Watson
GasConcentration = 107 PPM to IBM Watson
Humidity = 109% to IBM Watson
Temperature = 105 C to IBM Watson
GasConcentration = 107 PPM to IBM Watson
Humidity = 98% to IBM Watson
Temperature = 107 C to IBM Watson
GasConcentration = 103 PPM to IBM Watson
Humidity = 109% to IBM Watson
Temperature = 103 C to IBM Watson
GasConcentration = 102 PPM to IBM Watson
Humidity = 105% to IBM Watson
Temperature = 104 C to IBM Watson
GasConcentration = 107 PPM to IBM Watson
Humidity = 105% to IBM Watson
Temperature = 93 C to IBM Watson
GasConcentration = 108 PPM to IBM Watson
Humidity = 103% to IBM Watson
Temperature = 108 C to IBM Watson
GasConcentration = 100 PPM to IBM Watson
Humidity = 92% to IBM Watson
Temperature = 100 C to IBM Watson
GasConcentration = 90 PPM to IBM Watson
Humidity = 99% to IBM Watson
Temperature = 103 C to IBM Watson
GasConcentration = 107 PPM to IBM Watson
Humidity = 107% to IBM Watson
Temperature = 108 C to IBM Watson
GasConcentration = 97 PPM to IBM Watson
Humidity = 96% to IBM Watson
Temperature = 110 C to IBM Watson
GasConcentration = 92 PPM to IBM Watson
Humidity = 109% to IBM Watson
Temperature = 94 C to IBM Watson
```

WEB UI:



10:57



st.mybluemix.net

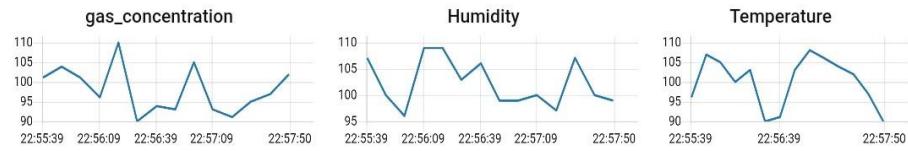


21



Smart Industry

Gas Detection



Smart Switch Board

ALARM OFF

ALARM ON



CLOUDANT:

The screenshot shows the Cloudant dashboard for a database named 'mydb'. The left sidebar contains navigation links: 'All Documents', 'Query', 'Permissions', 'Changes', and 'Design Documents'. The main content area is titled 'Document ID' and shows a table of documents. The table has three columns: 'id', 'key', and 'value'. The 'value' column contains JSON objects, each with a 'rev' field. The bottom of the screen shows a Windows taskbar with various application icons and system information like '85°F Haze' and '11:54 AM 18-Nov-22'.

id	key	value
0d910213621f26c419c01ddc60aee4d4	0d910213621f26c419c01ddc60aee4d4	{ "rev": "2-1e500b59f0093cd9799dba330c7f1..." }
0d910213621f26c419c01ddc60bd4fed	0d910213621f26c419c01ddc60bd4fed	{ "rev": "1-c5abb185a5ef517cd426f7309dea..." }
0d910213621f26c419c01ddc60d002fa	0d910213621f26c419c01ddc60d002fa	{ "rev": "1-84b7d9d3b16909e9e2c9a681022..." }
163b140cdd6c0710c4532159969105db	163b140cdd6c0710c4532159969105db	{ "rev": "1-61cd9af17f30a11c50952d533935..." }
1a96abc4136dc3acd540191c43641a35	1a96abc4136dc3acd540191c43641a35	{ "rev": "1-1c6154788545f245fbbad43c60f4..." }
1c93c65d488ff96833e85493972d1409	1c93c65d488ff96833e85493972d1409	{ "rev": "1-ec0c82c5476cece47b1445818cc1e..." }
1c93c65d488ff96833e85493972f95f0	1c93c65d488ff96833e85493972f95f0	{ "rev": "1-ddfe2b5215adde84ea60a77e9bda..." }
1c93c65d488ff96833e85493974b8caf	1c93c65d488ff96833e85493974b8caf	{ "rev": "1-c7d348f9f125a085a16614012191..." }
1c93c65d488ff96833e854939754e763	1c93c65d488ff96833e854939754e763	{ "rev": "1-503c12a35ecac3822a553f32dcc8..." }
1c93c65d488ff96833e8549397cfb571	1c93c65d488ff96833e8549397cfb571	{ "rev": "1-cab8ae88b2031c4615d5dd191db..." }
1c93c65d488ff96833e8549397d46ab5	1c93c65d488ff96833e8549397d46ab5	{ "rev": "1-57a0ff6c10470af99abb31a74ee..." }
1ea8bb627676e5fe989c080165ac46cb	1ea8bb627676e5fe989c080165ac46cb	{ "rev": "1-1c102e..." }

PYTHON CODE:

```
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=="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(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 IoT")
            time.sleep(10)
deviceCli.commandCallback=myCommandCallback
deviceCli.disconnect()

```

Reference link :

<https://node-red-kmnfb-2022-11-04.us-east.mybluemix.net/red/#flow/8e479358fc6b1ef9>

