SPRINT 1

Data Collection and Data Pre-processing

Date	12 November 2022
Team ID	PNT2022TMID27071
Project Name	Project - Gas Leakage Monitoring and Alerting System for Industries.

Data Collection:

- Data Collection is a process of gathering information from all the relevant sources to find a solution to the research problem.
- Most leak detectors are primarily responsible for locating the leak, determining the extend or rate of leakage, and keeping track of increase or decrease in leakage.

Pre-processing:

- Data pre-processing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure.
- Data can be cleaned by dividing it into equal size segment that are thus smoothed (binning), by fitting it to a linear or multiple regression function (regression), or by grouping it into cluster of similar data (clustering).

1.Introduction:

Internet of Things aim towards making life simpler by automating every small task around us. As much is IoT helping in automating tasks, the benefits of IoT can also be extended for enhancing the existing safety standards. Safety has always been an important criterion while designing home, buildings, industries as well as cities. The increased concentration of certain gases in the atmosphere can prove to be extremely dangerous. These gases might be flammable at certain temperature and humidity conditions, toxic after exceeding the specified concentrations limits or even a contributing factor in the air pollution of an area leading to problems such as smog and reduced visibility which can in turn cause severe accidents and also have adverse effect on the health of people.

Most of the societies have fire safety mechanism. But it can use after the fire exists. In order to have a control over such conditions we proposed system that uses sensors which is capable of detecting the gases such as LPG, CO2, CO and CH4.

This system will not only able to detect the leakage of gas but also alerting through audible alarms. Presence of excess amounts of harmful gases in environment then this system can notify the user. System can notify to society admin about the condition before mishap takes place through a message.

2. Problem Statement:

The purpose of this project is to detect the presence of LPG leakage as a part of a safety system and save the workers in the gas industries and causalities in the houses.

If the gas leaks, the sensor will send its data wirelessly to Arduino. Then, explosion prevention system will be activated. The system will turn the alarm/buzzer on, automatically releases gas regulator, and neutralizes the air with the exhaust fan and also intimate the user.

The gas leaked by an LPG cylinder if inhaled can lead to suffocation, as well as cause difficulty in walking or speaking. Your nervous system can get affected, while you can experience heart attack and rise in your blood pressure. Hence, it is important to be careful if you detect a LPY cylinder leak.

A gas detector can sound an alarm to operators in the area where the leak is occurring, giving them the opportunity to leave. This type of device is important because there are many gases that can be harmful to organic life, such as humans or animals.

3. Proposed Solution:

The purpose of this project is to detect the presence of LPG leakage as a part of a safety system and save the workers in the gas industries and causalities in the houses.

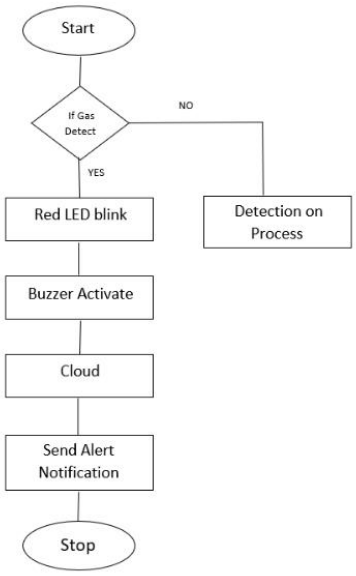
If the gas leaks, the sensor will send its data wirelessly to Arduino. Then, explosion prevention system will be activated.

The system will turn the alarm/buzzer on, automatically releases gas regulator, and neutralizes the air with the exhaust fan and also intimate the user.

4. Theoretical Analysis:

4.1 Block Diagram:

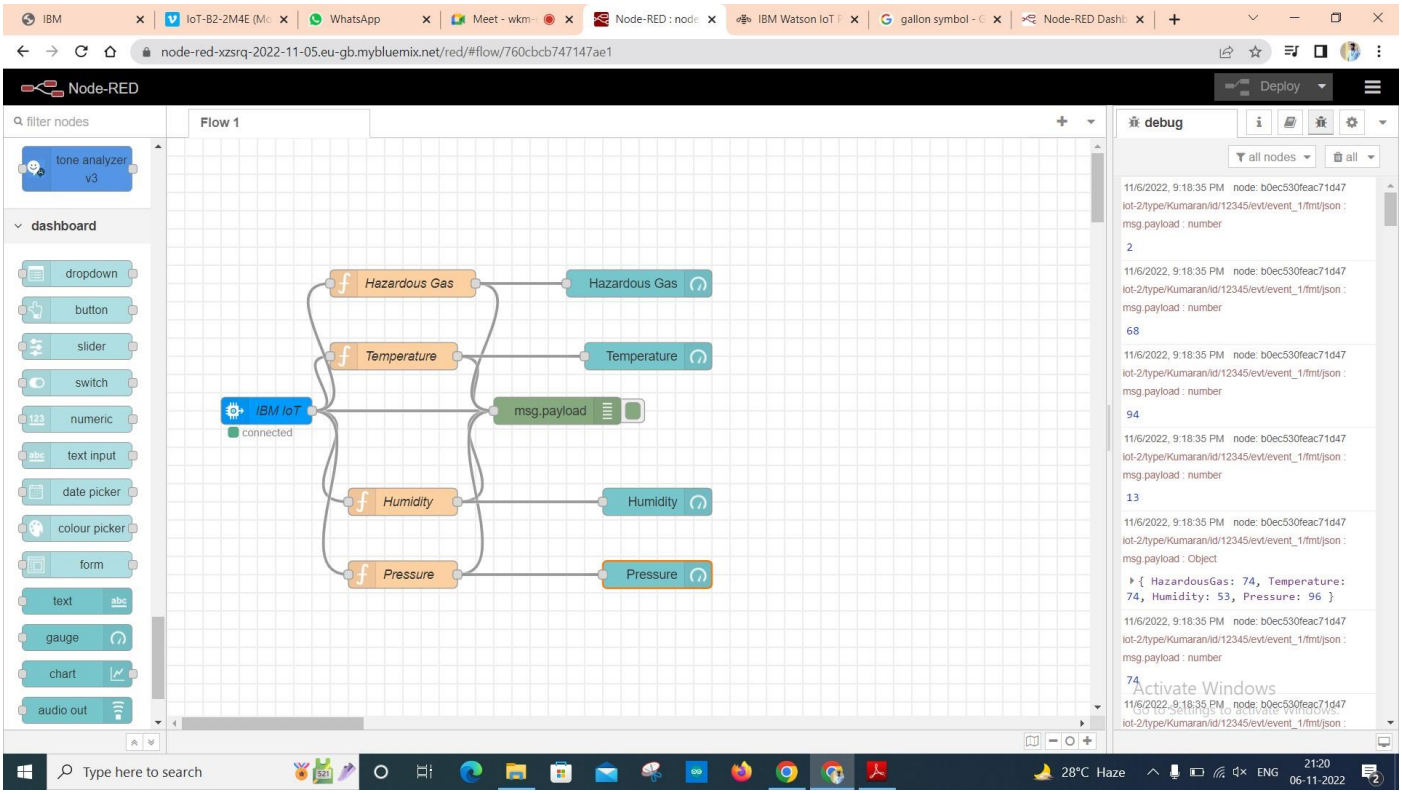
In order to implement the solution, the following approach as shown in the block diagram is used,



4.2 Required Software Installation:

4.2.A Node-Red:

Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things. Node-RED provides a web browser-based flow editor, which can be used to create JavaScript functions.



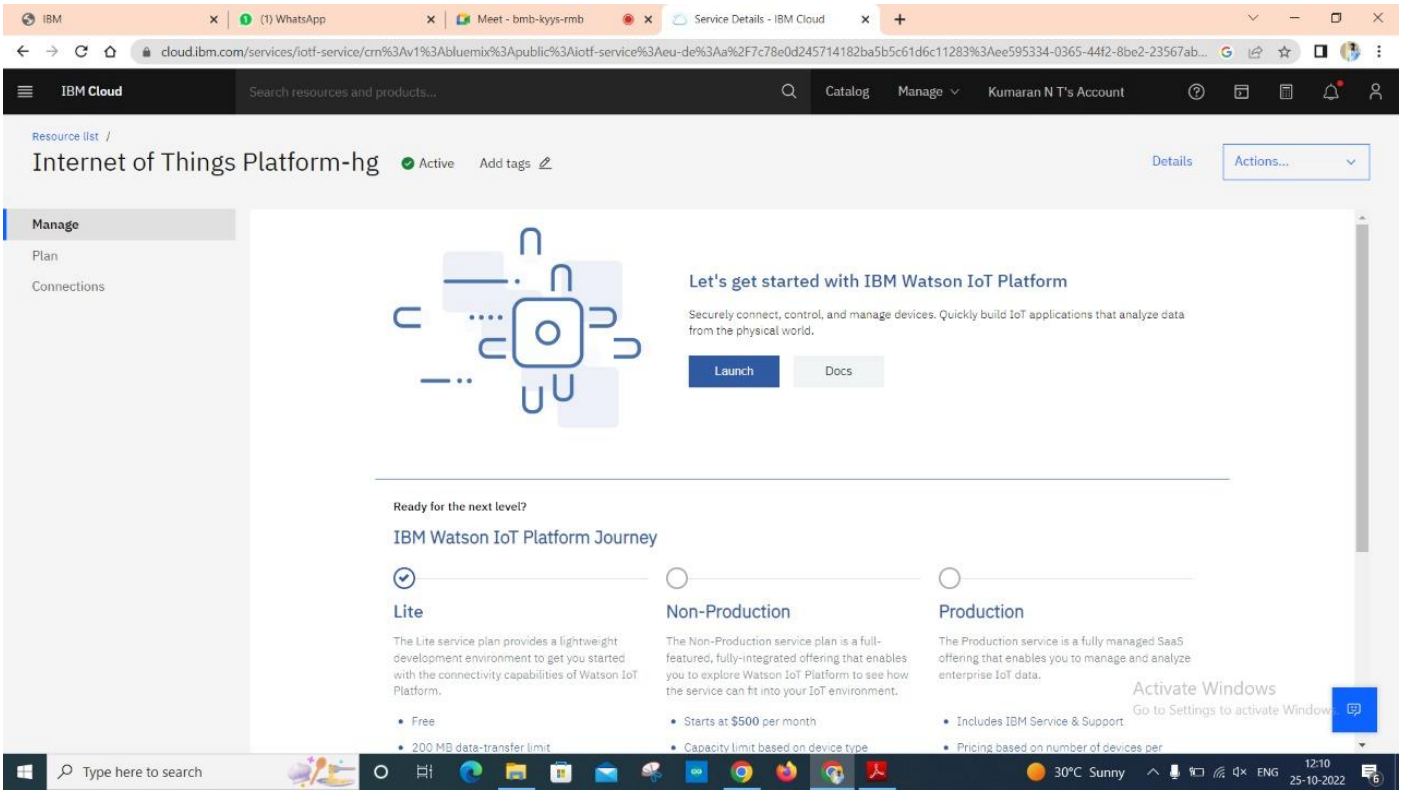
Installation of IBM IoT and Dashboard nodes for Node-Red:

In order to connect to IBM Watson IoT platform and create the Web App UI these nodes are required,

1. IBM IoT node
2. Dashboard node

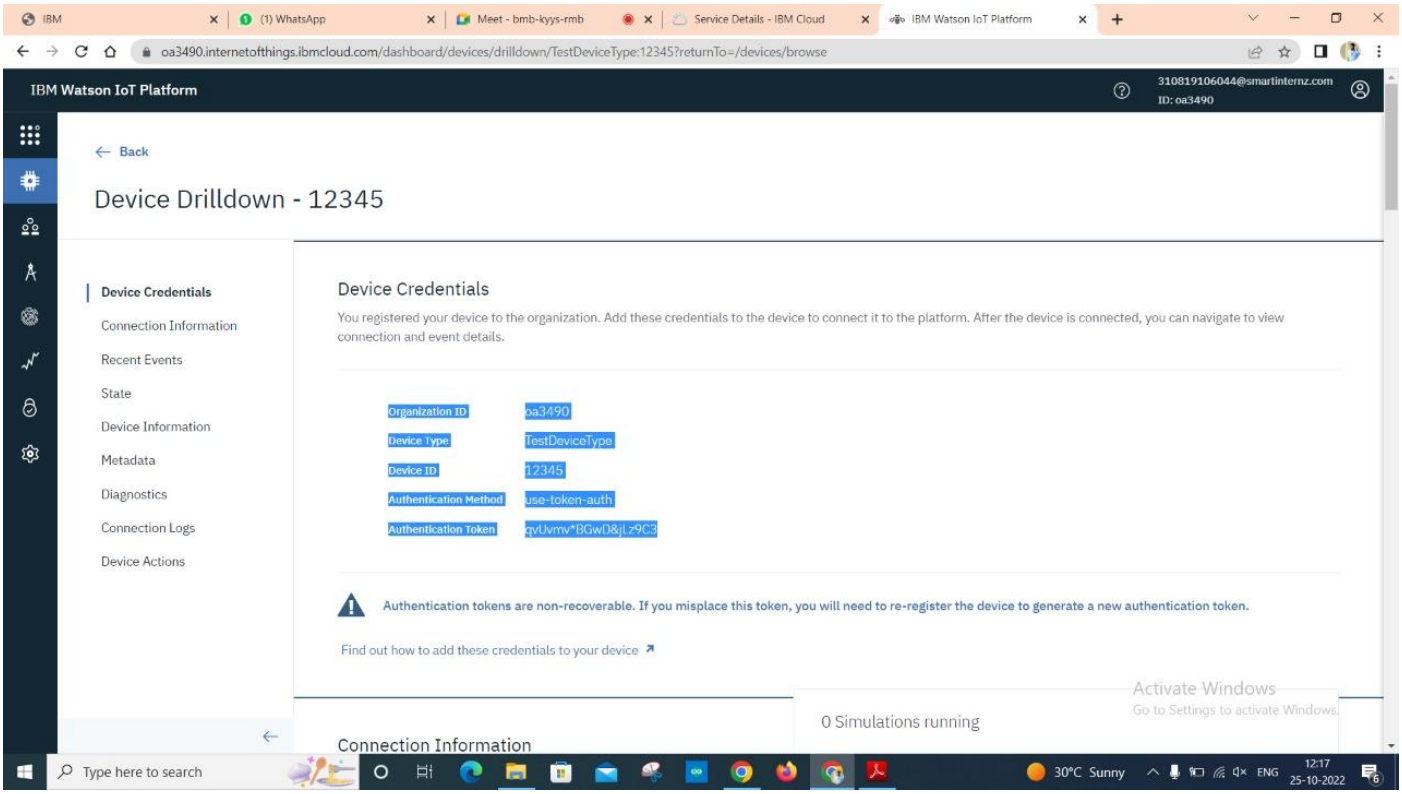
4.2.B IBM Watson IoT Platform:

A fully managed, cloud-hosted service with capabilities for device registration, connectivity, control, rapid visualization and data storage. IBM Watson IoT Platform is a managed, cloud-hosted service designed to make it simple to derive value from your IoT devices.



Steps to configure:

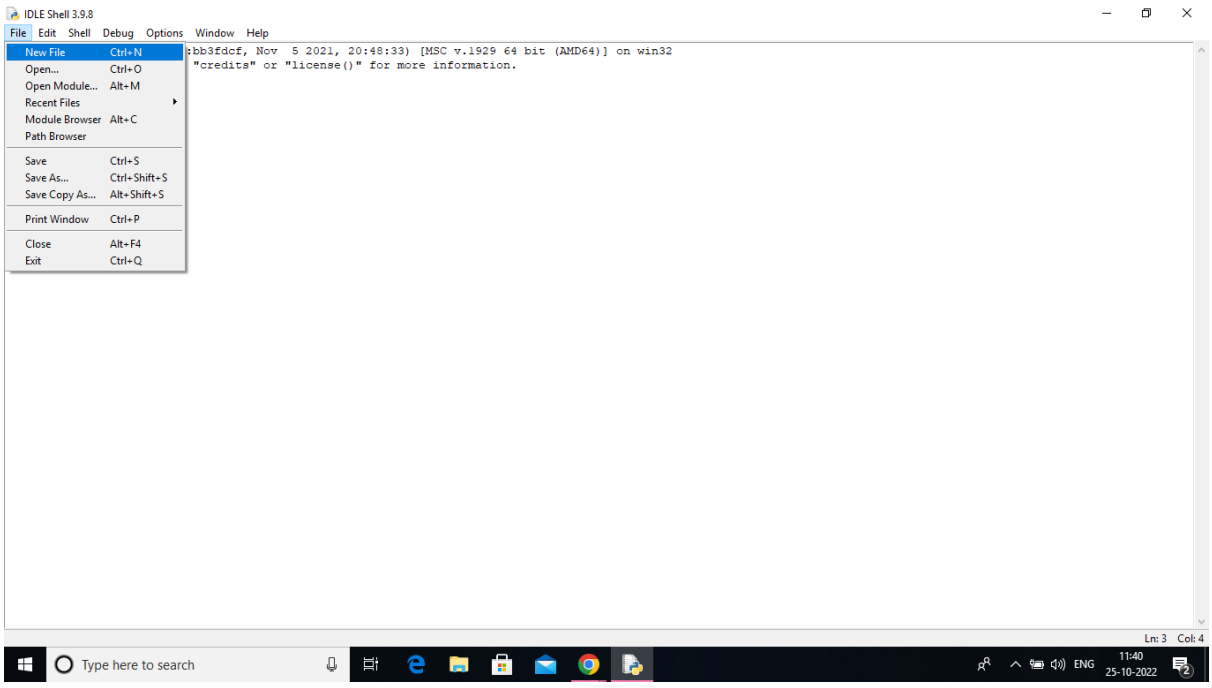
- Create an account in IBM cloud using your email ID
- Create IBM Watson Platform in services in your IBM cloud account
- Launch the IBM Watson IoT Platform
- Create a new device
- Give credentials like device type, device ID, Auth. Token
- Create API key and store API key and token elsewhere.



4.2.C Python IDE

Install Python3 compiler

Install any python IDE to execute python scripts, in my case I used Spyder to execute the code.



Code:

```
#IBM Watson IOT Platform
#pip install wiotp-sdk
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "yf0dyy",
        "typeId": "Kumaran",
        "deviceId":"12345"
    },
    "auth": {
        "token": "VJTDPRX@f&4Vuox8ms"
    }
}

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(0,100)
    hum=random.randint(0,100)
    pre=random.randint(0,100)
    haz=random.randint(0,100)

    myData={'Temperature':temp,
            'Humidity':hum,
            'Pressure':pre,
            'HazardousGas':haz
            }
    client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0, onPublish=None)
    print("Published data Successfully: %s", myData)
    if(haz>90):
        print("Exhaust Fan is ON")
    else:
        print("Exhaust Fan is OFF")
    client.commandCallback = myCommandCallback
    time.sleep(2)
client.disconnect()
```

4.3 IoT Simulator

In our project in the place of sensors we are going to use IoT sensor simulator which give random readings to the connected cloud.

The link to simulator:

https://watson-iot-sensor-simulator.mybluemix.net/

We need to give the credentials of the created device in IBM Watson IoT Platform to connect cloud to simulator.