

SPRINT 2

Model Creation

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

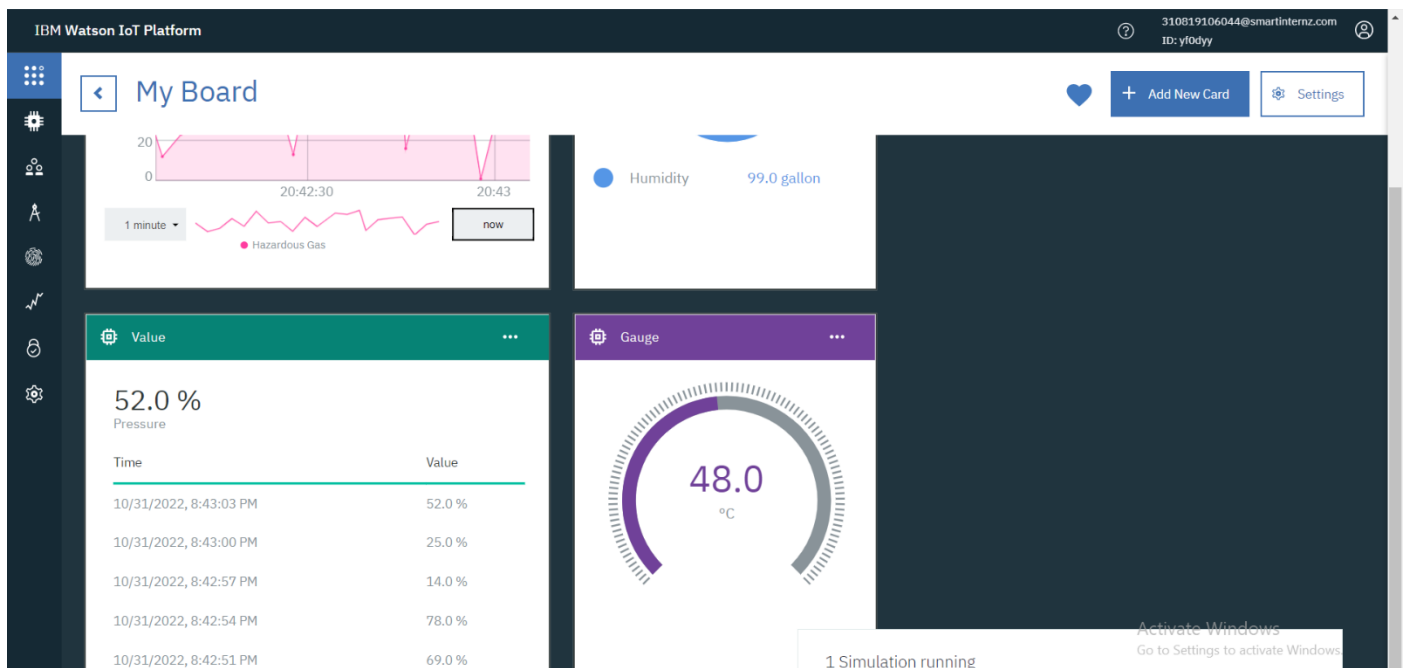
Model Creation:

- Pre-processed data are sent to the cloud in order to be processed, analysed, or modelled in order to build smart applications. Because IoT devices usually don't emit data at regular intervals, IoT time series data is highly irregular regarding their sampling rate within as well as across devices.
 - ✓ Name: - Gas Leakage Monitoring and Alerting System for Industries.
 - ✓ Device Type: Detecting the leakage of the gas in industry
 - ✓ Kind: Sensor
 - ✓ Producer: By the rate of leakage
 - ✓ Frequency: Every time when the leakage is detected.

5. Building Project:

5.1 Connecting IoT Simulator to IBM Watson IoT Platform:

- Open link provided in above section 4.3
- Give the credentials of your device in IBM Watson IoT Platform
- Click on connect
- My credentials given to simulator are:
 - ✓ Organization ID: oa3490
 - ✓ Device Type: TestDeviceType
 - ✓ Device ID: 12345
 - ✓ Authentication Method: use-token-auth
 - ✓ Authentication Token: qvUvmv*BGwD&jLz9C3



You can see the received data in graphs by creating cards in Boards tab

- You will receive the simulator data in cloud
- You can see the received data in Recent Events under your device
- Data received in this format(json)

IBM Watson IoT Platform

310819106044@smartinternz.com
ID: yf0dy

Browse Action Device Types Interfaces

Add Device

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Disconnected	Kumaran	Device	Oct 31, 2022 11:38 AM	

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

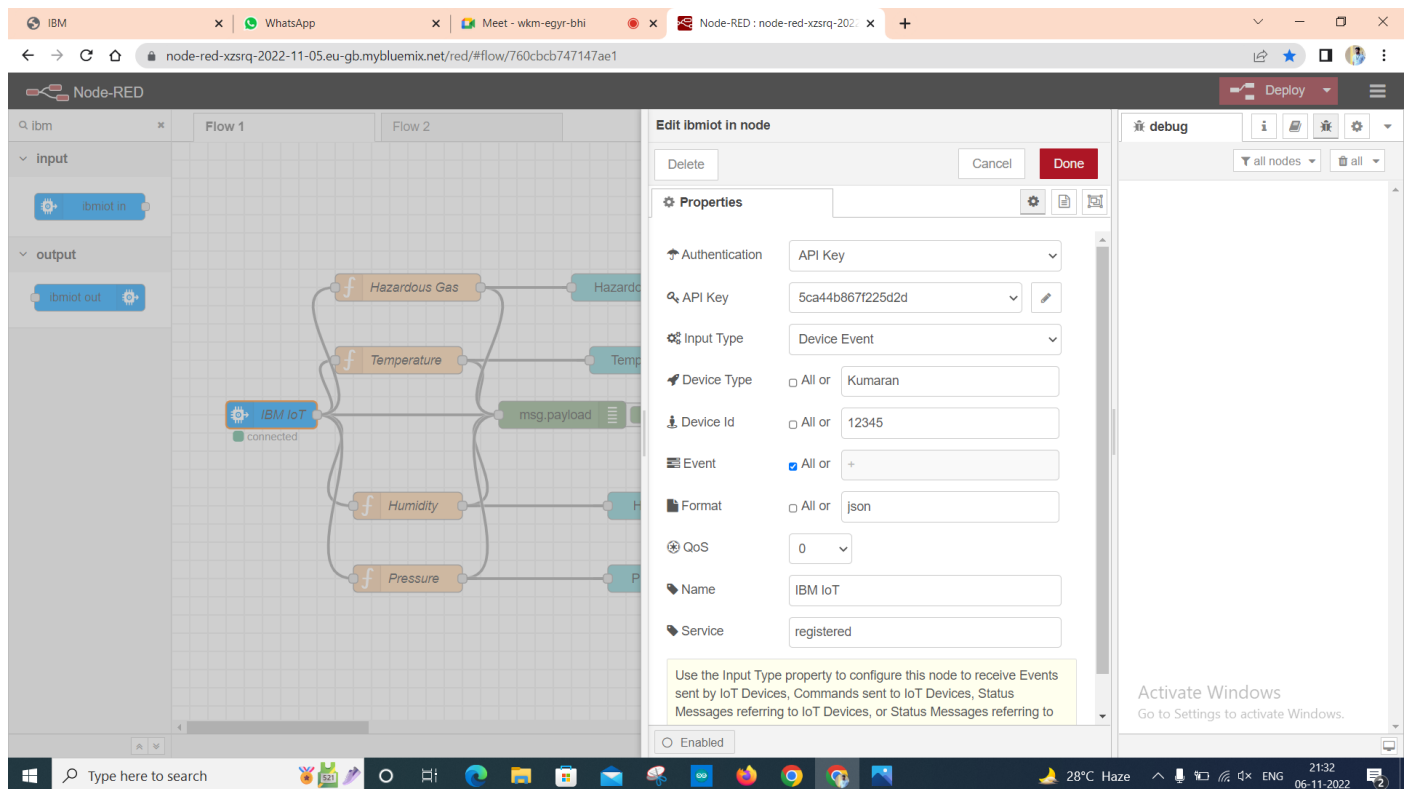
Event	Value	Format	Last Received
event_1	{"Hazardous Gas":61,"Temperature":88,"Humidit...	json	a few seconds ago
event_1	{"Hazardous Gas":20,"Temperature":36,"Humidit...	json	a few seconds ago
event_1	{"Hazardous Gas":79,"Temperature":56,"Humidit...	json	a few seconds ago
event_1	{"Hazardous Gas":52,"Temperature":82,"Humidit...	json	a few seconds ago
event_1	{"Hazardous Gas":26,"Temperature":33,"Humidit...	json	a few seconds ago

1 Simulation running

Activate Windows
Go to Settings to activate Windows.

5.2 Configuration of Node-Red to collect IBM cloud data:

The node IBM IoT App In is added to Node-Red workflow. Then the appropriate device credentials obtained earlier are entered into the node to connect and fetch device telemetry to Node-Red.



Once it is connected Node-Red receives data from the device

Display the data using debug node for verification

Connect function node and write the Java script code to get each reading separately.

The Java script code for the function node is

- ✓ Type `msg.payload=msg.payload.Temperature`
- ✓ Type `msg.payload=msg.payload.Humidity`
- ✓ Type `msg.payload=msg.payload.HazardousGas`
- ✓ Type `msg.payload=msg.payload.d.Pressure`

Finally connect Gauge nodes from dashboard to see the data in UI

The screenshot displays the Node-RED web interface in a browser. The top bar shows several open tabs, including 'Node-RED : node' and 'IBM Watson IoT'. The address bar indicates the URL: `node-red-xzsrq-2022-11-05.eu-gb.mybluemix.net/red/#flow/760cbcb747147ae1`.

The main workspace, titled 'Flow 1', contains a flow starting with an 'IBM IoT' node (labeled 'connected'). This node is connected to four function nodes: 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. Each function node is connected to a corresponding gauge node on the right: 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. A 'msg.payload' node is also connected to the function nodes.

On the left, a 'dashboard' panel is visible, containing various UI components like 'dropdown', 'button', 'slider', 'switch', 'numeric', 'text input', 'date picker', 'colour picker', 'form', 'text', 'gauge', 'chart', and 'audio out'. The 'gauge' component is highlighted.

On the right, a 'debug' console shows a log of messages. The messages are JSON objects with the following structure:

```
{
  "time": "11/6/2022, 9:18:35 PM",
  "node": "b0ec530feac71d47",
  "iot-2/type/KumaranId/12345/evt/event_1/fmt/json": {
    "msg.payload": "number"
  }
}
```

The log shows multiple messages, with the last one displaying a JSON object: `{ HazardousGas: 74, Temperature: 74, Humidity: 53, Pressure: 96 }`.