

SPRINT 2

Model Creation

Date	15 November 2022
Team ID	PNT2022TMID27080
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:

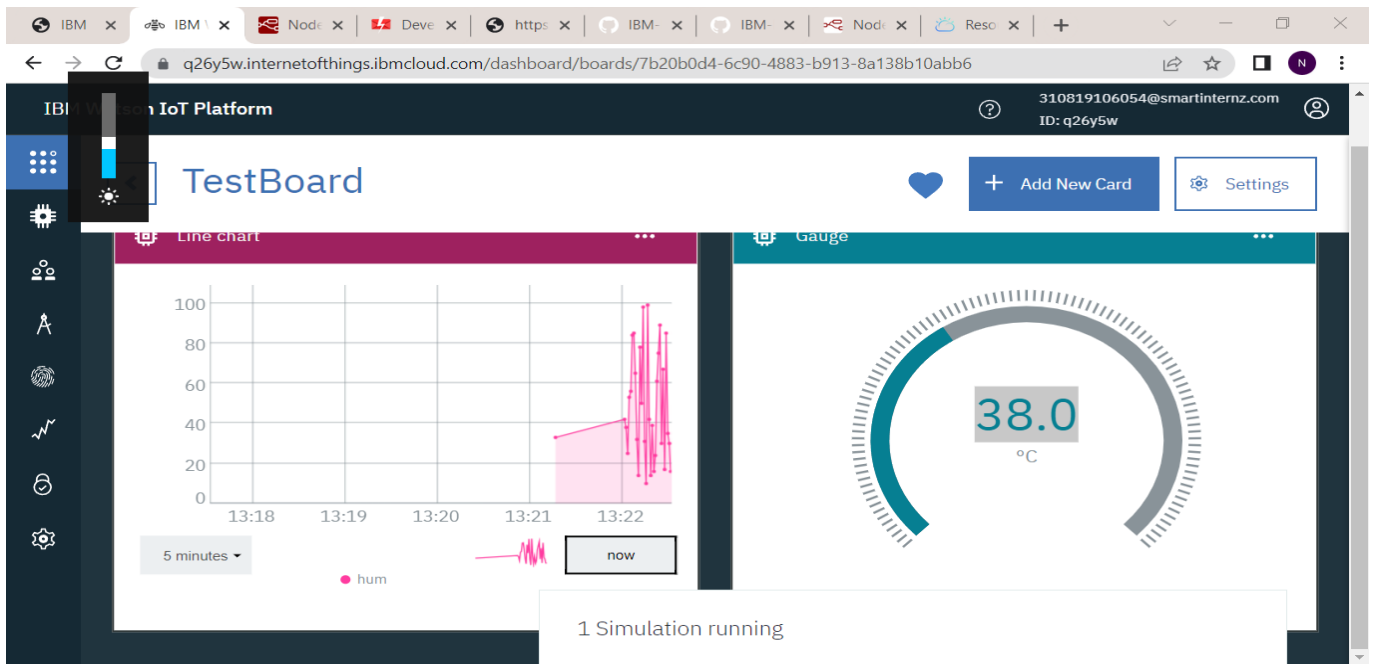
Organization ID: q26y5w

Device Type: TestDeviceType

Device ID: 2022

Authentication Method: use-token-auth

Authentication Token: uu(rZQw9292EfSFGDX



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

310819106054@smartinternz.com
ID: q26y5w

Browse Action Device Types Interfaces

Add Device

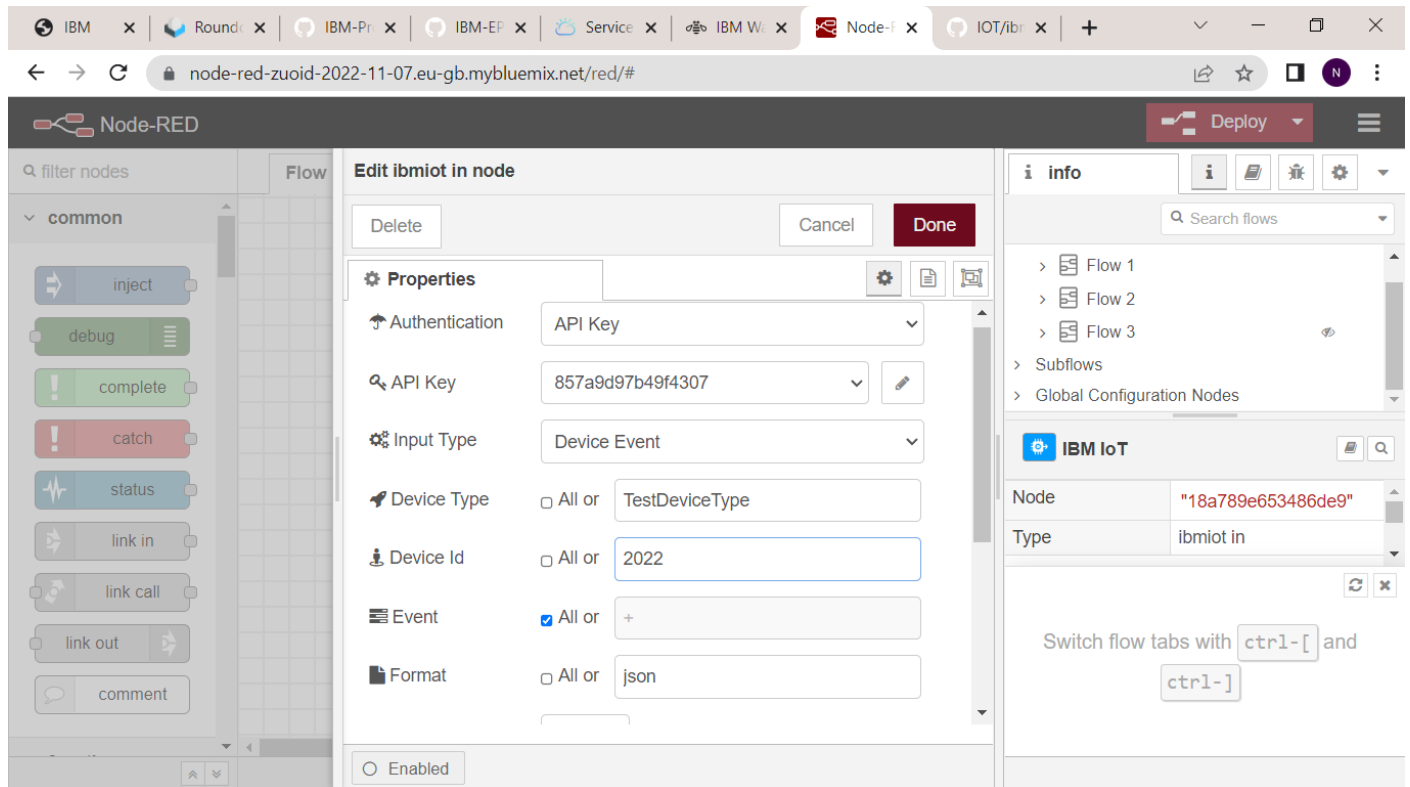
The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
event_test	{"temp":96,"hum":8}	json	a few seconds ago
event_test	{"temp":71,"hum":21}	json	a few seconds ago
event_test	{"temp":83,"hum":10}	json	a few seconds ago
event_test	{"temp":56,"hum":52}	json	a few seconds ago
event_test	{"temp":48,"hum":1}	json	a few seconds ago

1 Simulation running

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

IBM | Webma | IBM-EPI | IBM-EPI | Service | IBM W | Noc x | IOT/ibm | Node-R | Node-R | +

node-red-zuoid-2022-11-07.eu-gb.mybluemix.net/red/#flow/ba844783534d964f

Node-RED

Deploy

Flow 1 | Flow 2

input

- ibmiot in

output

- ibmiot out

Flow 1 Diagram:

```
graph LR; IBMIoT[IBM IoT] --> HazardousGas; IBMIoT --> Temperature; IBMIoT --> Humidity; IBMIoT --> Pressure; HazardousGas --> HazardousGasOut[HazardousGas]; Temperature --> TemperatureOut[Temperature]; Humidity --> HumidityOut[Humidity]; Pressure --> PressureOut[Pressure];
```

debug

all nodes

msg.payload : Object

```
{ temperature: 47, humidity: 5, Hazardousgas: 93, pressure: 37 }
```

11/9/2022, 9:17:58 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : number
93

11/9/2022, 9:17:58 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : number
47

11/9/2022, 9:17:58 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : number
5

11/9/2022, 9:17:58 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : number