

Gas Leakage monitoring & Alerting system for Industries

SPRINT - 2

TEAM ID	PNT2022TMID04012
DATE	14 TH NOVEMBER 2022

IBM Watson IoT platform, Workflows for IoT scenarios using Node-red:

Building Project:

Connecting IOT Simulator to IBM Watson IOTPlatform

Give the credentials of your device in IBM Watson

"orgId": "wv4o8f",

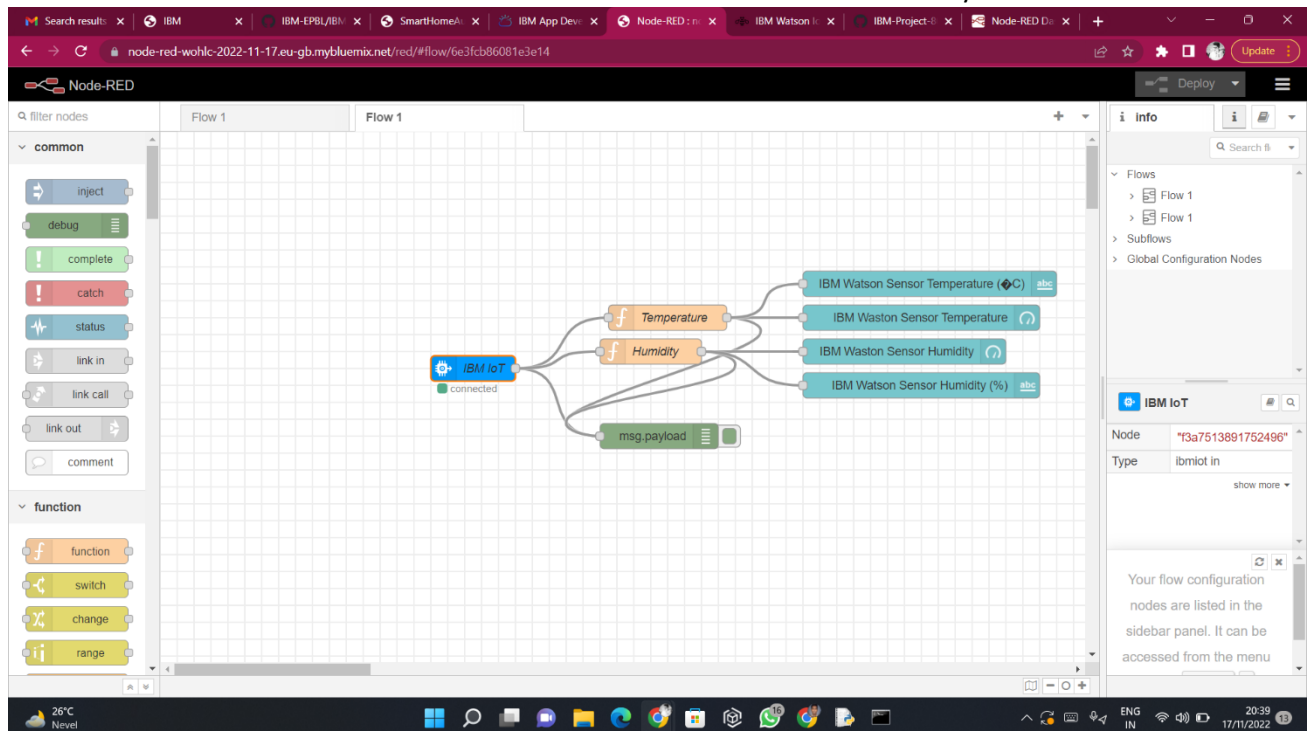
The screenshot displays a development environment with two main windows. The left window is a terminal running a Python script that simulates an IoT device, publishing temperature and humidity data. The right window is a code editor showing the Python code for the simulator. The code includes the following details:

- Terminal Output:** Shows the execution of the Python script, including the installation of the `wiotp-sdk` and the successful connection of the device. It lists several published data points, such as: `Published data Successfully: {'temperature': 1, 'humidity': 35}`.
- Code Editor:** Contains the Python code for the simulator. Key parts include:
 - Imports: `import random`, `import time`, and `from wiotp.sdk.device import DeviceClient`.
 - Configuration: `myConfig = {'identity': {'orgId': 'wv4o8f', 'typeId': 'NodeMCU', 'deviceId': '12345'}, 'auth': {'token': '123456789'}}`.
 - Client Initialization: `client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)` and `client.connect()`.
 - Simulation Loop: A `while True:` loop that generates random temperature and humidity values, publishes them as JSON events, and includes a `time.sleep(2)` delay.

"typeId": "NodeMCU",

"deviceId": "12345"

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



The Node-Red also receive data from the Open Weather API by HTTP GET request. An inject trigger is added to perform HTTP request for every certain interval. HTTP request node is configured with URL we saved before.

