

## Creating a Node-Red Web Application to view data in Separate Numerical form

Date	07 November 2022
Team ID	PNT2022TMID04619
Project Name	Project - Gas Leakage Monitoring and Alerting System for Industries.

➤ In IBM cloud dashboard, click on Cloud Foundry apps

The screenshot displays the IBM Cloud dashboard interface. The top navigation bar includes the IBM Cloud logo, a search bar, and user account information. The main content area is titled 'Resource list' and features a table with columns for Name, Group, Location, Product, Status, and Tags. A 'Create resource' button is visible in the top right corner. The table lists a single resource: 'Node RED XZSRQ 2022-11-05' under the group 'Gas Leakage Monitoring / Kumaran', located in 'London', with the product 'Node.js' and status 'Started'. A sidebar on the left shows various service categories like Compute, Containers, Networking, Storage, AI / Machine Learning, Analytics, Blockchain, Databases, Developer tools, Logging and monitoring, and Migration. An 'Activate Windows' watermark is present in the bottom right corner of the dashboard area.

Name	Group	Location	Product	Status	Tags
Node RED XZSRQ 2022-11-05	Gas Leakage Monitoring / Kumaran	London	Node.js	Started	-

- A new window appears where we need to NODE-RED SELDZ app created before.

Resource list

Name	Group	Location	Product	Status	Tags
Node RED XZSRQ 2022-11-05	Gas Leakage Monitoring / Kumaran	London	Node.js	Started	-

Activate Windows  
Go to Settings to activate Windows

- Click on Visit App URL in Node RED SELDZ service dashboard.

Node RED XZSRQ 2022-11-05 Running [Visit App URL](#) [Add tags](#) [Details](#) [Actions...](#)

Getting started

Overview

Runtime

Connections

Logs

API Management

Autoscaling

Instances

Health

100%

1/1 instance(s) are running

MB memory per instance

0 2048 256

Runtime

Node.js

256

Total MB allocation

1.75 GB still available

Free Used

Runtime cost

Current and estimated cost excludes connected services.

\$0.00 \$0.00

Estimated total for billing period

Connections (1)

node-red-fdefl-2022--cloudant-1666683139018-10339

Activate Windows  
Go to Settings to activate Windows

- Click on your Node-RED flow editor where you will be redirected to the Node-RED flow editor.

The screenshot shows a web browser window with multiple tabs. The active tab is 'Node-RED on IBM Cloud' with the URL 'node-red-xzsrq-2022-11-05.eu-gb.mybluemix.net'. The page has a dark red header with 'Node-RED on IBM Cloud' and a large red banner with 'Node-RED' and 'Flow-based programming for the Internet of Things'. Below the banner, there is a text block describing Node-RED as a programming tool for wiring together hardware devices, APIs and online services. A button labeled 'Go to your Node-RED flow editor' is present. Below the button, there is a link 'Learn how to customise Node-RED'. At the bottom right, there is a 'Activate Windows' watermark.

Node-RED on IBM Cloud

# Node-RED

Flow-based programming for the Internet of Things

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

More information about Node-RED, including documentation, can be found at [nodered.org](https://nodered.org).

[Go to your Node-RED flow editor](#)

[Learn how to customise Node-RED](#)

Activate Windows  
Go to Settings to activate Windows.

## Customising your instance of Node-RED

The screenshot shows the Node-RED flow editor interface. The browser window has tabs for 'WhatsApp', 'IBM', 'IoT-B2-2M4E (Morning Session)', 'SmartHomeAutomationusingIBM', and 'Node-RED'. The active tab is 'Node-RED' with the URL '127.0.0.1:1880/#flow/b47948623bf1c79f'. The interface is divided into three main sections: a left sidebar with a 'filter nodes' search bar and two categories of nodes ('common' and 'function'), a central workspace with a grid and a 'Flow 1' tab, and a right sidebar with an 'info' panel showing flow details and a 'Deploy' button. The 'common' nodes include inject, debug, complete, catch, status, link in, link call, link out, and comment. The 'function' nodes include function, switch, change, and range. The 'info' panel shows the flow name 'Flow 1' and its ID 'b47948623bf1c79f'. At the bottom, there is a Windows taskbar with various icons and a system tray showing '25°C Cloudy' and the date '03-11-2022'.

Node-RED

filter nodes

Flow 1

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch
- change
- range

Deploy

info

Search flows

Flows

- Flow 1

Subflows

Global Configuration Nodes

Flow 1

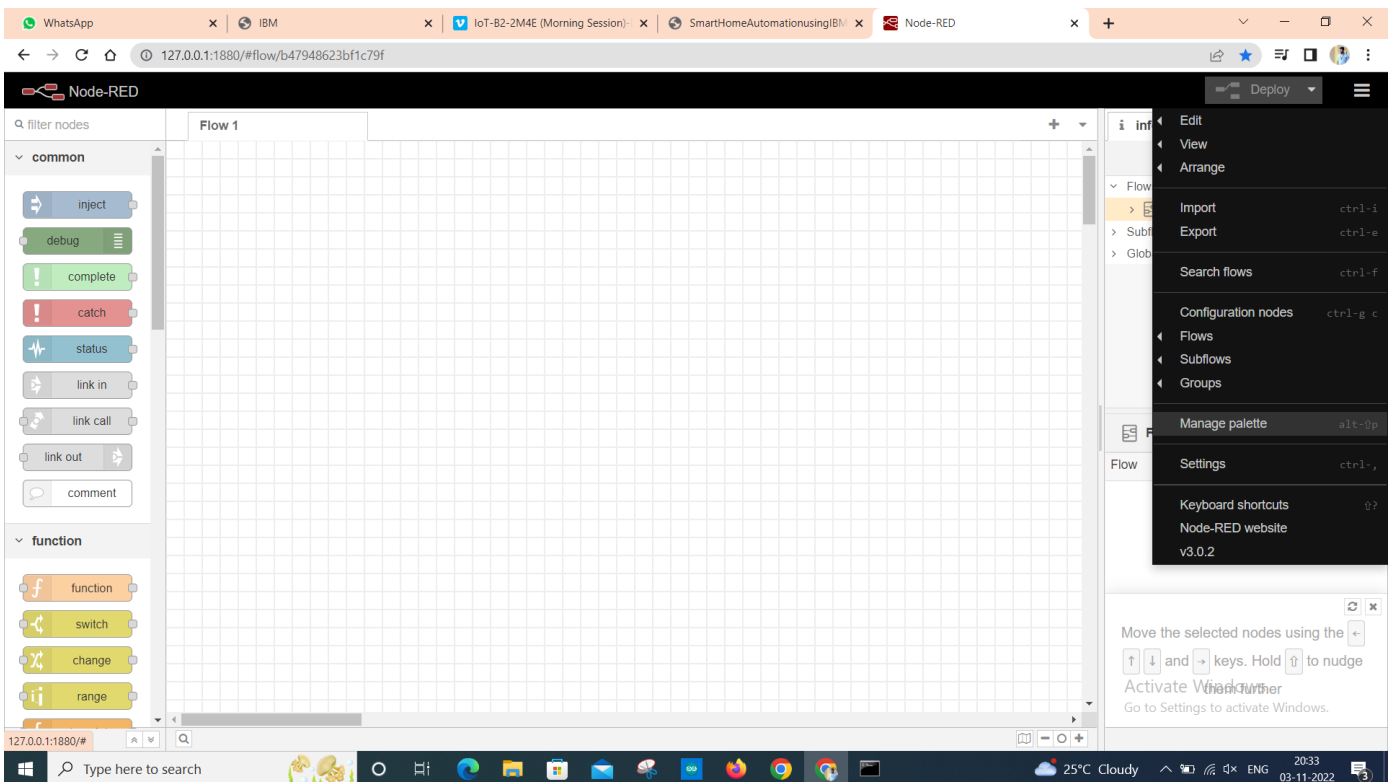
Flow

"b47948623bf1c79f"

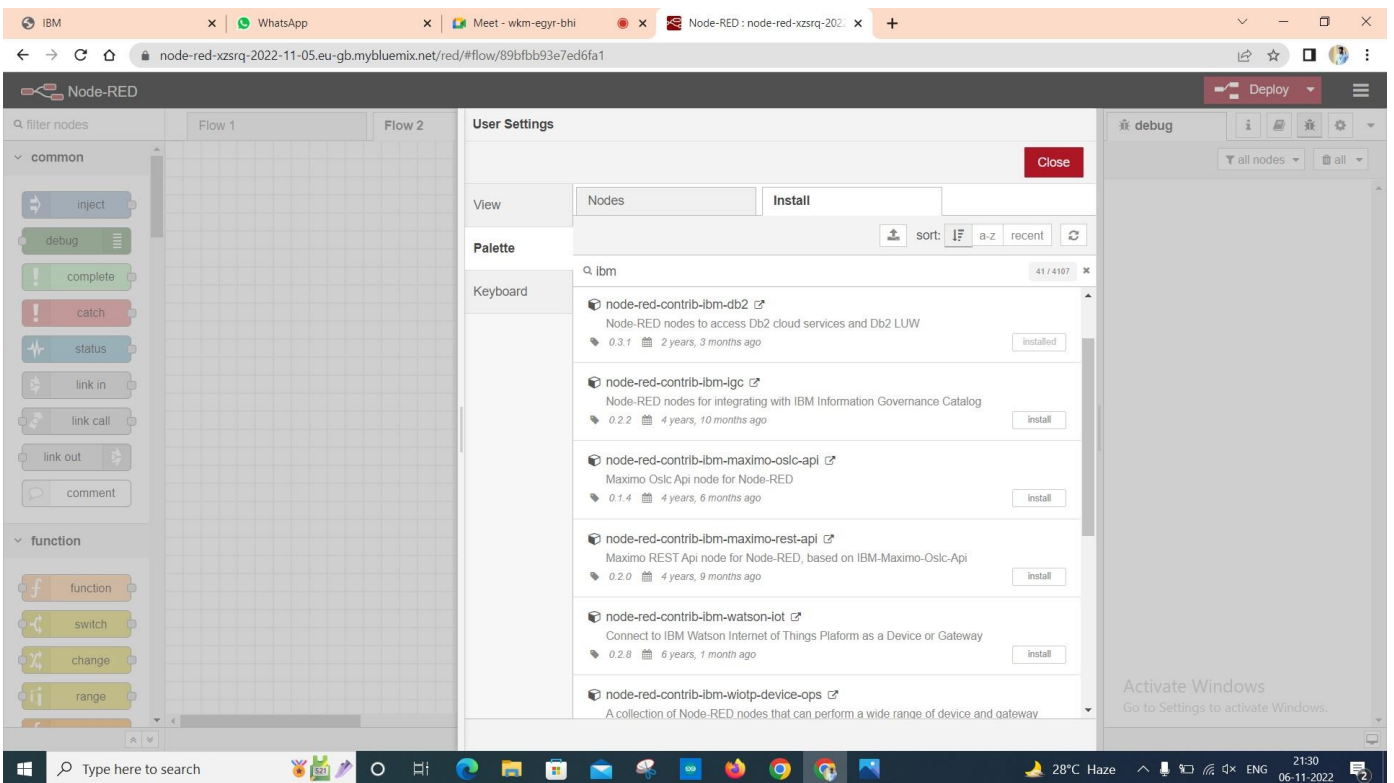
Move the selected nodes using the ↑ ↓ ← → keys. Hold ⇧ to nudge

Activate Windows  
Go to Settings to activate Windows.

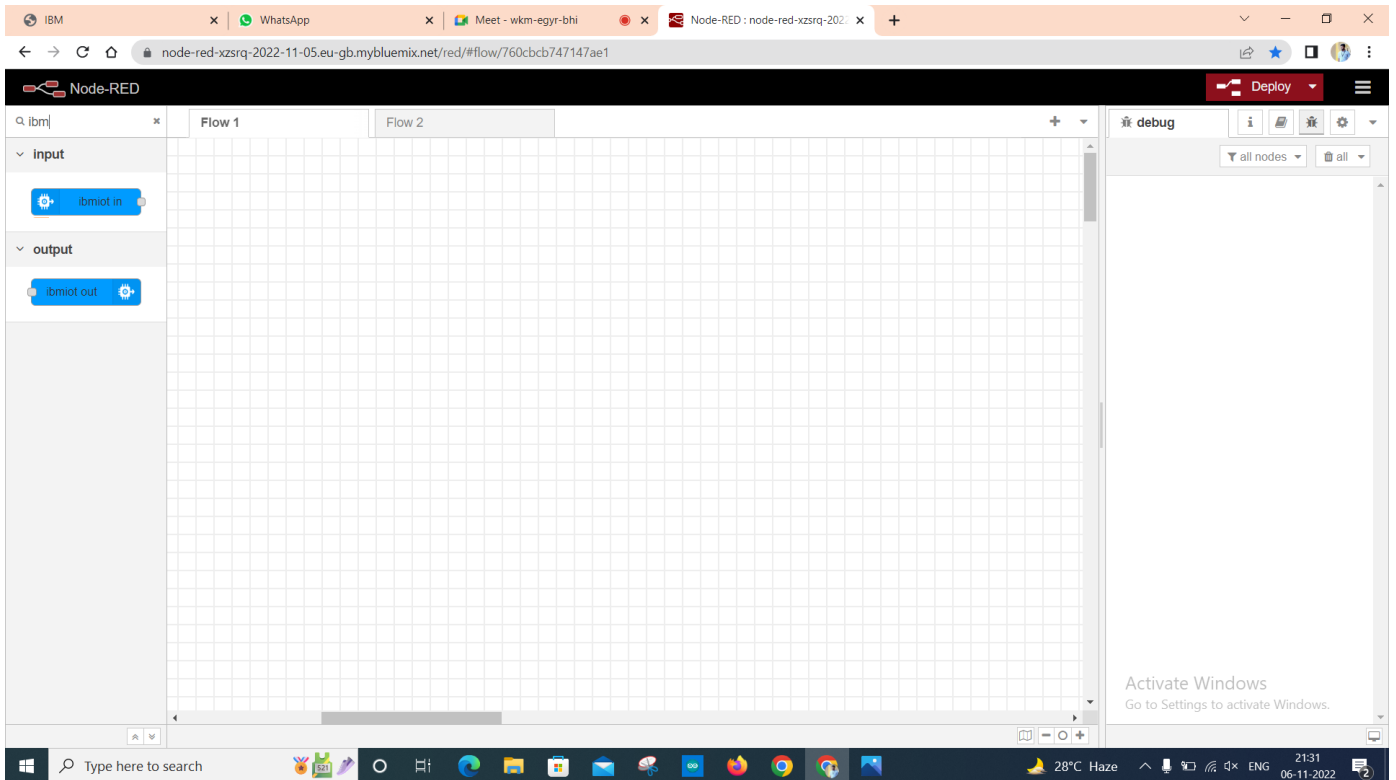
- To install IBM nodes in Node-red flow editor click on manage palette in the menu option which is on the top-right of the screen.



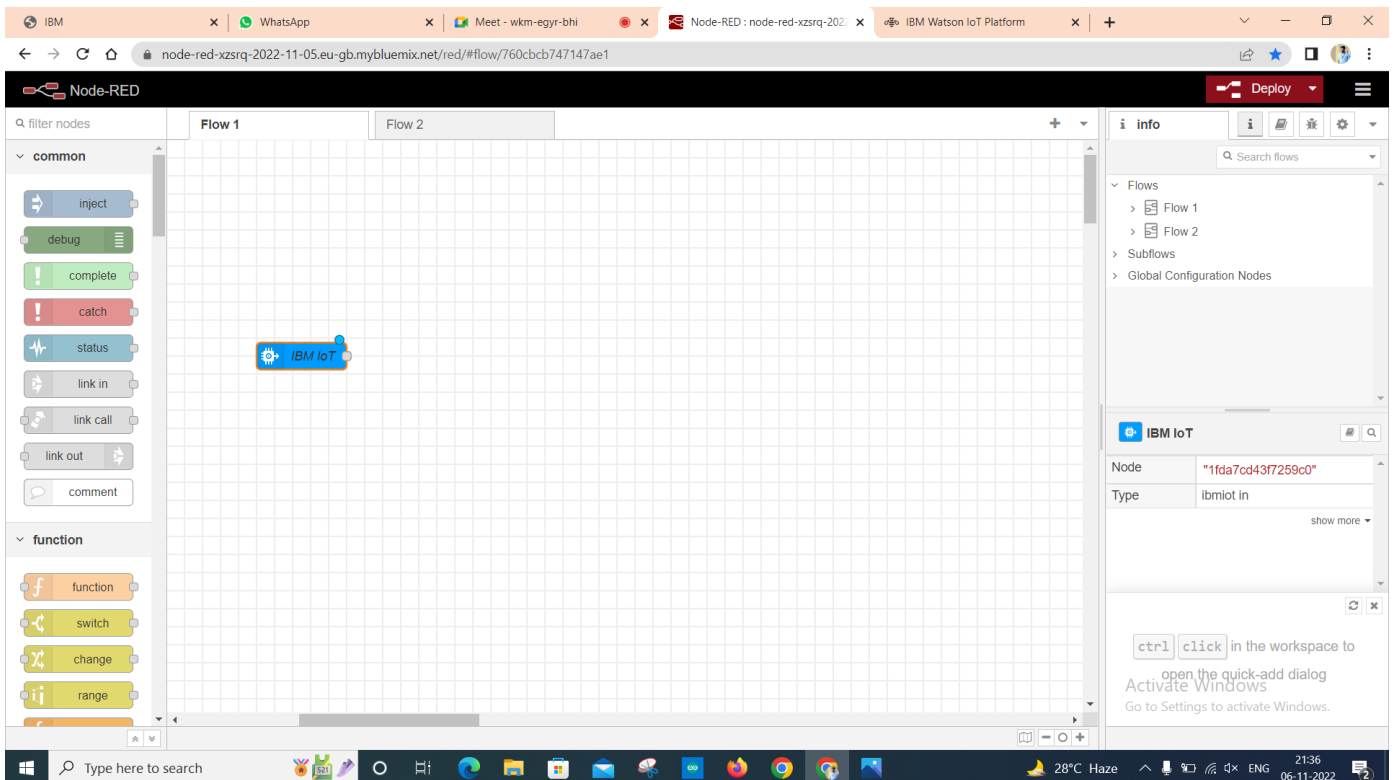
- In install section search for ibmiot and install the ibm nodes to flow editor.



➤ Search for IBM nodes in the filter nodes section



➤ To Retrieve the data from the IBM IoT platform by using Node-RED IBM IoT Input node and double click on the IBM IoT input node



- Select API Key from Authentication in properties.
- In API Key paste API Key, API Token and server name and update it

**Edit ibmiot in node**

Delete Cancel Done

**Properties**

- Authentication: API Key
- API Key: 5ca44b867f225d2d
- Input Type: Device Event
- Device Type: All or Kumaran
- Device Id: All or 12345
- Event: All or +
- Format: All or json
- QoS: 0
- Name: IBM IoT
- Service: registered

Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to

Enabled

**info**

Search flows

- Flows
  - Flow 1
  - Flow 2
- Subflows
- Global Configuration Nodes

**IBM IoT**

Node	"c9195d4cbebc02e8"
Type	ibmiot in

show more

ctrl click in the workspace to open the quick-add dialog  
Activate Windows  
Go to Settings to activate Windows.

- Also update your input type as event, Device type, Device ID, command and format in the properties section and click on Done

- To generate API Key go to IBM IoT platform
- In Apps Section -> Click on Generate API Key

The screenshot shows the IBM Watson IoT Platform dashboard. The user is logged in as '310819106044@smartinternz.com' with ID 'yf0ddy'. The 'Generate API Key' button is visible in the top right. Below it, a table lists API keys. The first key, 'a-yf0ddy-iwy9pm96o', is selected, and its details are shown in the 'API Key Information' section.

Key	Description	Role	Expires
a-yf0ddy-iwy9pm96o	API Key for the device simulator	Standard Application	-
a-yf0ddy-tbwwm8i7z2	-	Standard Application	-

The 'API Key Information' section for the selected key shows:

Key	a-yf0ddy-iwy9pm96o	Last Edited By	-
Description	API Key for the device simulator	Expires	Never
Date Added	Nov 6, 2022 9:33 PM		
Last Update	Nov 6, 2022 9:33 PM		

A status bar at the bottom indicates '1 Simulation running'.

- Click on Deploy option to check the connection status. If the status is disconnected check for IBM IoT properties and try again.

The screenshot shows the Node-RED interface. A flow named 'Flow 1' contains an 'IBM IoT' node (labeled 'connected') connected to a 'msg.payload' node. The 'Deploy' button is visible in the top right. The debug console on the right shows a series of messages from the 'IBM IoT' node, each containing a timestamp, node ID, and a JSON payload with sensor data.

```

msg payload: Object
  Hazardous Gas: 98, Temperature: 96, Humidity: 83, Pressure: 100 }

11/6/2022, 8:29:05 PM node:b0ec530feac71d47
lot-2?type/Kumaran/Id/12345/ev/1/fmt/json:
msg payload: Object
  Hazardous Gas: 37, Temperature: 13, Humidity: 18 }

11/6/2022, 8:29:08 PM node:b0ec530feac71d47
lot-2?type/Kumaran/Id/12345/ev/1/fmt/json:
msg payload: Object
  Hazardous Gas: 18, Temperature: 59, Humidity: 20, Pressure: 60 }

11/6/2022, 8:29:11 PM node:b0ec530feac71d47
lot-2?type/Kumaran/Id/12345/ev/1/fmt/json:
msg payload: Object
  Hazardous Gas: 65, Temperature: 83, Humidity: 98, Pressure: 52 }

11/6/2022, 8:29:14 PM node:b0ec530feac71d47
lot-2?type/Kumaran/Id/12345/ev/1/fmt/json:
msg payload: Object
  Hazardous Gas: 83, Temperature: 74, Humidity: 83, Pressure: 73 }

11/6/2022, 8:29:17 PM node:b0ec530feac71d47
lot-2?type/Kumaran/Id/12345/ev/1/fmt/json:
msg payload: Object
  Hazardous Gas: 100, Temperature: 10, Humidity: 0, Pressure: 87 }
  
```



- Place the debug node in the flow editor and click on deploy to see the temperature and humidity value in the debug tab

The screenshot shows the Node-RED web interface in a browser. The flow editor displays a flow named 'Flow 1' with two nodes: 'IBM IoT' (connected) and 'msg.payload'. The left sidebar shows the 'common' and 'function' node palettes. The right sidebar shows the 'debug' tab, which displays a list of messages. A red box highlights a portion of the debug console showing the following messages:

```
msg.payload: Object
{
  Hazardous Gas: 98, Temperature:
96, Humidity: 83, Pressure: 100 }
11/6/2022, 8:29:05 PM node: b0ec530fac71d47
msg.payload: Object
{
  Hazardous Gas: 37, Temperature:
13, Humidity: 83, Pressure: 18 }
11/6/2022, 8:29:08 PM node: b0ec530fac71d47
11/6/2022, 8:29:11 PM node: b0ec530fac71d47
msg.payload: Object
{
  Hazardous Gas: 18, Temperature:
59, Humidity: 20, Pressure: 60 }
11/6/2022, 8:29:14 PM node: b0ec530fac71d47
11/6/2022, 8:29:17 PM node: b0ec530fac71d47
msg.payload: Object
{
  Hazardous Gas: 65, Temperature:
83, Humidity: 98, Pressure: 52 }
11/6/2022, 8:29:14 PM node: b0ec530fac71d47
11/6/2022, 8:29:17 PM node: b0ec530fac71d47
msg.payload: Object
{
  Hazardous Gas: 83, Temperature:
74, Humidity: 83, Pressure: 73 }
11/6/2022, 8:29:17 PM node: b0ec530fac71d47
11/6/2022, 8:29:17 PM node: b0ec530fac71d47
msg.payload: Object
{
  Hazardous Gas: 100, Temperature:
10, Humidity: 0, Pressure: 87 }
```

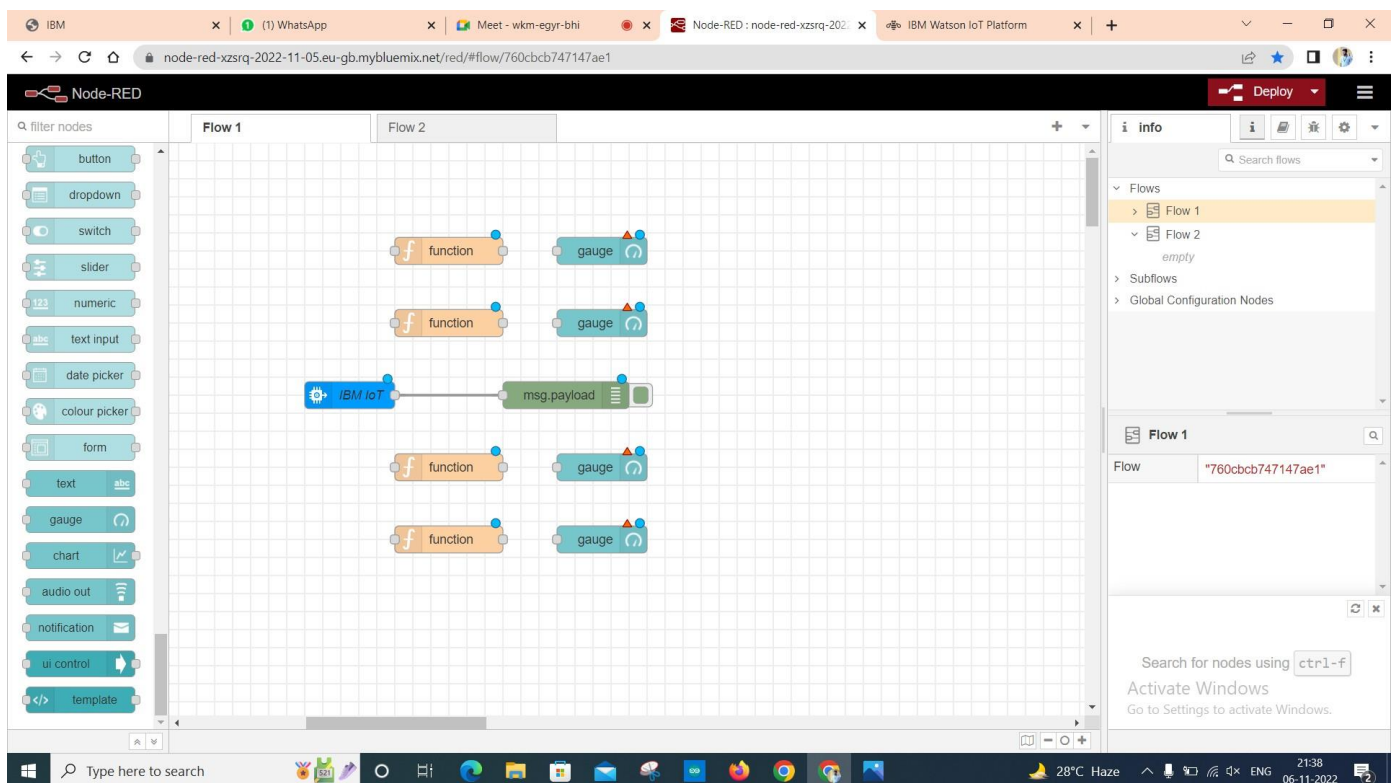
- Install the dashboard node from the manage pallet to create a UI to display temperature and humidity values in the Dashboard

The screenshot shows the Node-RED web interface with the 'User Settings' dialog box open. The 'Nodes' tab is selected, and the 'Palette' section shows a search for 'dashboard'. The following nodes are listed in the palette:

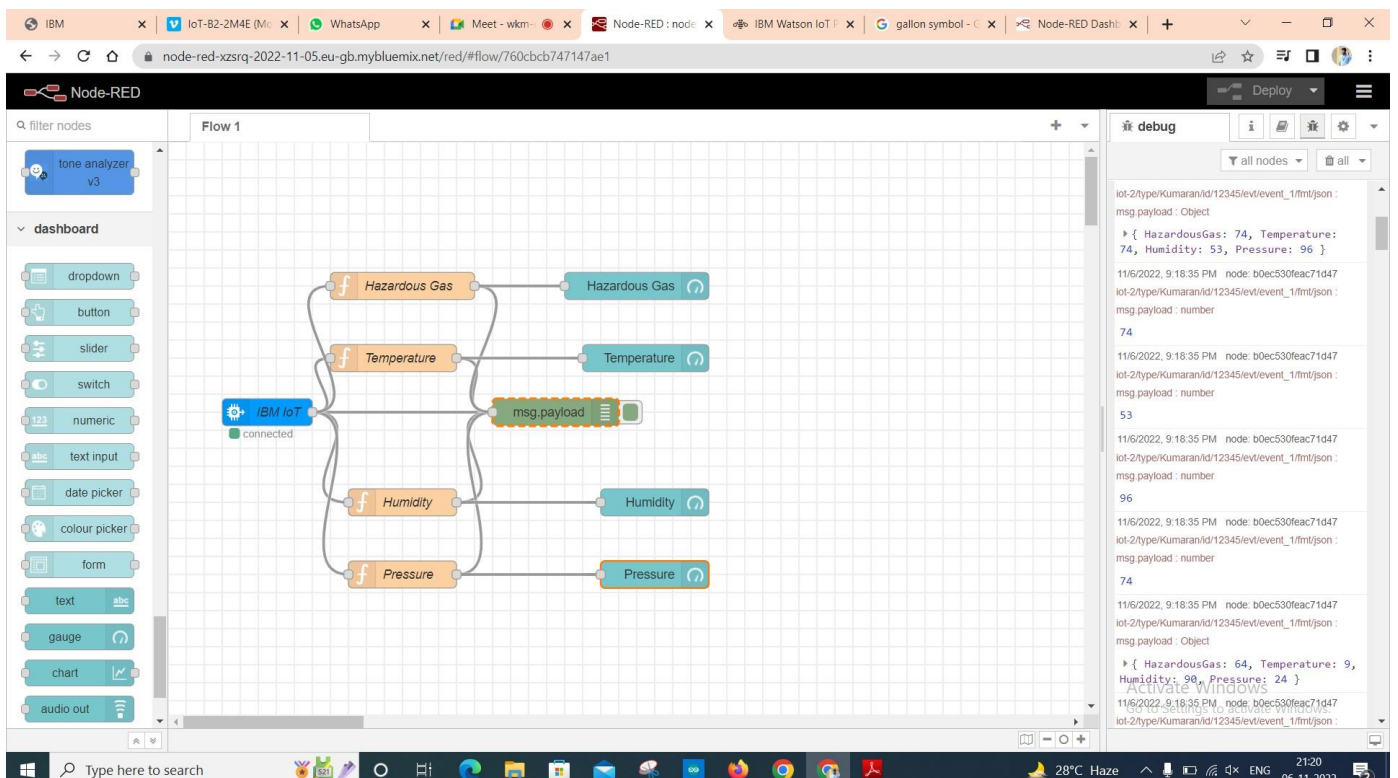
- node-red-dashboard**: A set of dashboard nodes for Node-RED. Version 3.2.0, 2 months ago. **Installed**
- feezal**: Web Components based Dashboard UI with WYSIWYG Editor. Version 0.8.1, 1 year, 11 months ago. **Install**
- node-red-contrib-dashboard-average-bars**: Calculate and display the average values of msg.payload in a bar chart. Version 0.0.6, 4 years, 4 months ago. **Install**
- node-red-contrib-dashboard-bar-chart-data**: Transforms sensor measurements or meter readings to be displayed in dashboard-chart as bar-chart. Version 0.9.8, 10 months ago. **Install**
- node-red-contrib-dashboard-sum-bars**: Calculate and display the sum of msg.payload in a bar chart. Version 0.0.1, 4 years, 4 months ago. **Install**



- Drag and place the function node and gauge node in the flow editor to separate the temperature and humidity value



- Double click on function and update the details as follow,
- Type `msg.payload=msg.payload.Temperature` in one function.
- Type `msg.payload=msg.payload.Humidity` in another function
- Type `msg.payload=msg.payload.HazardousGas`
- Type `msg.payload=msg.payload.d.Pressure`
- To separate the humidity and temperature values from payload and click deploy



- Select gauge function and these nodes to temperature, pressure, hazardous gas and humidity

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1'. It starts with an 'IBM IoT' node (blue) connected to four function nodes (orange): 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. Each function node is connected to a corresponding gauge node (blue): 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. A 'msg.payload' node (green) is also connected to the function nodes. The left sidebar shows the 'filter nodes' search bar and a list of nodes under the 'dashboard' category, including 'gauge'. The right sidebar shows the 'debug' console with a list of messages and their payloads.

- Edit temperature, hazardous gas, pressure and humidity nodes and deploy it.

The screenshot shows the Node-RED web interface with the 'Edit gauge node' dialog box open for the 'Hazardous Gas' gauge. The dialog box has a 'Delete' button, 'Cancel' and 'Done' buttons, and a 'Properties' tab. The properties include: 'Group' ( '[Hazardous Gas] Gas Leakage' ), 'Size' ( 'auto' ), 'Type' ( 'Gauge' ), 'Label' ( 'Hazardous Gas' ), 'Value format' ( '{{value}}' ), 'Units' ( 'ppm' ), 'Range' ( 'min 0 max 100' ), 'Colour gradient' ( a color bar with green, yellow, and red ), 'Sectors' ( '0 ... optional ... optional ... 100' ), 'Class' ( 'Optional CSS class name(s) for widget' ), and 'Name' ( an empty field ). The 'Enabled' checkbox is checked. The background shows the same flow as the previous screenshot, but the 'Hazardous Gas' gauge node is highlighted.

➤ After editing the nodes, deploy it

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow named 'Flow 1'. On the left, a sidebar contains a 'filter nodes' search bar and a 'dashboard' panel with various widgets like dropdown, button, slider, switch, numeric, text input, date picker, colour picker, form, text, gauge, chart, and audio out. The central workspace features a flow starting with an 'IBM IoT' node (labeled 'connected'). This node branches into five parallel paths, each passing through a function node (labeled 'f') and then to an output node. The function nodes are labeled 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. The output nodes are also labeled 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. A 'msg.payload' node is also present in the flow. On the right, a 'debug' console shows the data flow, including JSON objects for 'HazardousGas', 'Temperature', 'Humidity', and 'Pressure'.

```
iot-2/type/Kumaran/Id/12345/evt/event_1/fmt/json :  
msg.payload : Object  
  { HazardousGas: 74, Temperature:  
    74, Humidity: 53, Pressure: 96 }  
11/6/2022, 9:18:35 PM node: b0ec530feac71d47  
iot-2/type/Kumaran/Id/12345/evt/event_1/fmt/json :  
msg.payload : number  
74  
11/6/2022, 9:18:35 PM node: b0ec530feac71d47  
iot-2/type/Kumaran/Id/12345/evt/event_1/fmt/json :  
msg.payload : number  
53  
11/6/2022, 9:18:35 PM node: b0ec530feac71d47  
iot-2/type/Kumaran/Id/12345/evt/event_1/fmt/json :  
msg.payload : number  
96  
11/6/2022, 9:18:35 PM node: b0ec530feac71d47  
iot-2/type/Kumaran/Id/12345/evt/event_1/fmt/json :  
msg.payload : number  
74  
11/6/2022, 9:18:35 PM node: b0ec530feac71d47  
iot-2/type/Kumaran/Id/12345/evt/event_1/fmt/json :  
msg.payload : Object  
  { HazardousGas: 64, Temperature: 9,  
    Humidity: 90, Pressure: 24 }  
11/6/2022, 9:18:35 PM node: b0ec530feac71d47  
iot-2/type/Kumaran/Id/12345/evt/event_1/fmt/json :
```

## RESULT:

Thus, the Node-Red Web Application is created successfully.