

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

Date	11 November 2022
Team ID	PNT2022TMID18989
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 'Resource list' dashboard. The browser's address bar shows 'cloud.ibm.com/resources'. The dashboard header includes the IBM Cloud logo, a search bar, and navigation links for 'Catalog', 'Manage', and the user account 'Kumaran N T's Account'. A 'Create resource' button is visible in the top right corner.

The main content area is titled 'Resource list' and features a table with the following columns: Name, Group, Location, Product, Status, and Tags. The table lists resources categorized by type on the left sidebar:

- Compute (1): Node RED XZSRQ 2022-11-05 (Group: Gas Leakage Monitoring / Kumaran, Location: London, Product: Node.js, Status: Started)
- Containers (0)
- Networking (0)
- Storage (0)
- AI / Machine Learning (0)
- Analytics (0)
- Blockchain (0)
- Databases (2+)
- Developer tools (13+)
- Logging and monitoring (0)
- Migration (0)

An 'Activate Windows' watermark is visible in the bottom right corner of the dashboard area. The Windows taskbar at the bottom shows the search bar and various application icons, with the system clock indicating 21:48 on 06-11-2022.

- 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](#)

Getting started

Overview

Health

100%

1/1 instance(s) are running

MB memory per instance

0 2048 256

Runtime cost

Current and estimated cost excludes connected services.

\$0.00 \$0.00

Estimated total for billing period

Runtime

Node.js

256

Total MB allocation

1.75 GB still available

Free Used

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 includes a left sidebar with a 'filter nodes' search bar and two categories of nodes: 'common' (inject, debug, complete, catch, status, link in, link call, link out, comment) and 'function' (function, switch, change, range). The main workspace is a grid labeled 'Flow 1'. The right sidebar shows a tree view with 'Flows' (Flow 1), 'Subflows', and 'Global Configuration Nodes'. Below the tree view, there is a 'Flow 1' section with a 'Flow' label and a value 'b47948623bf1c79f'. At the bottom, there is a 'Deploy' button and a 'Move the selected nodes using the' section with instructions on using arrow keys and a 'Check out' button.

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

Flows

- Flow 1
- Subflows
- Global Configuration Nodes

Flow 1

Flow

"b47948623bf1c79f"

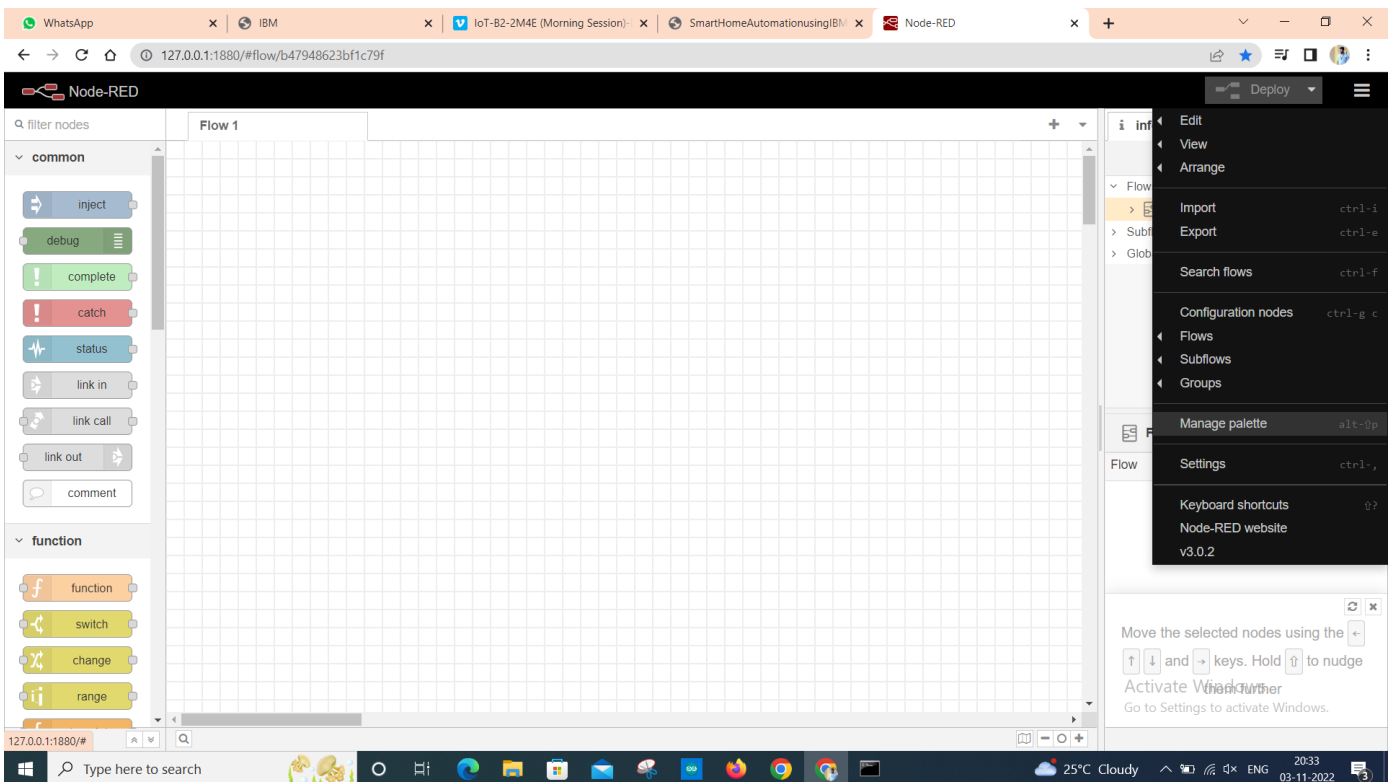
Move the selected nodes using the

↑ ↓ ← → keys. Hold ⇧ to nudge

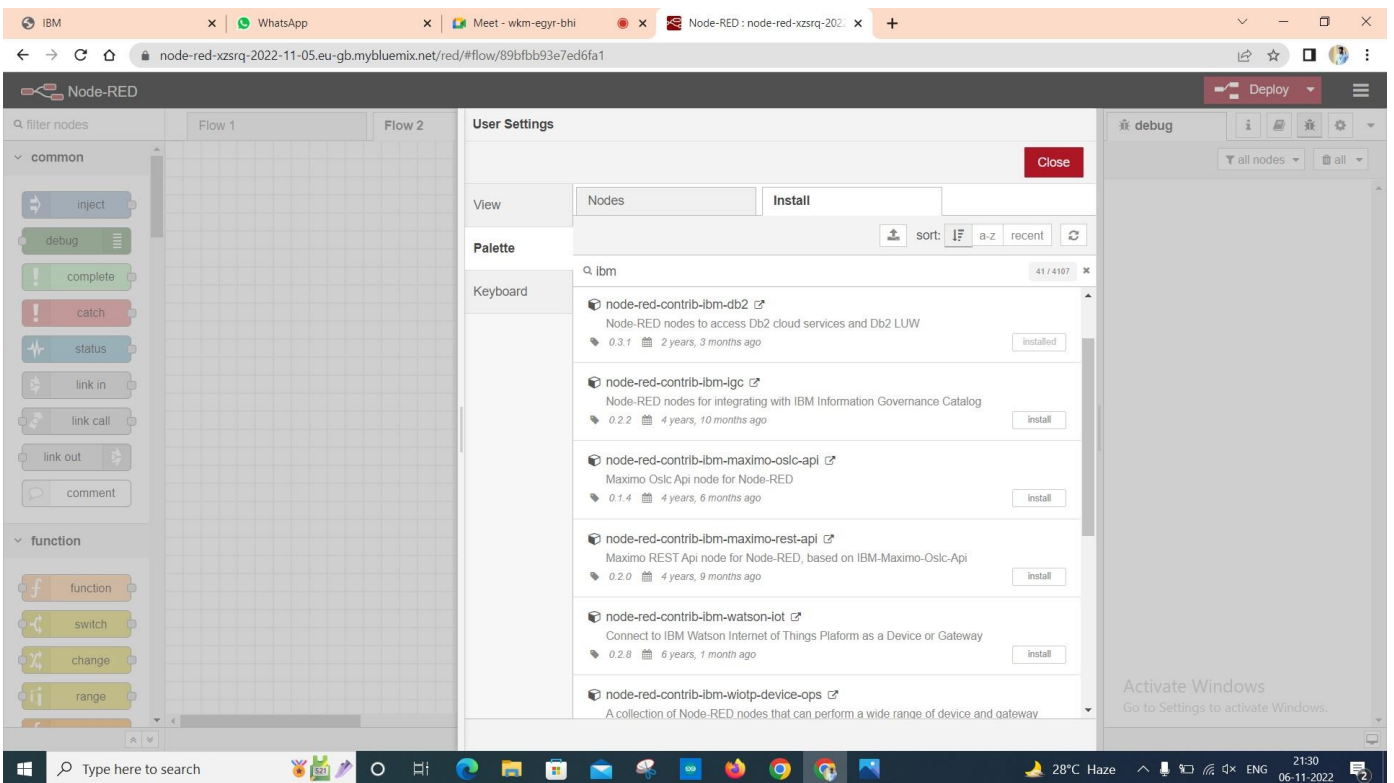
Activate Windows

Go to Settings to activate Windows.

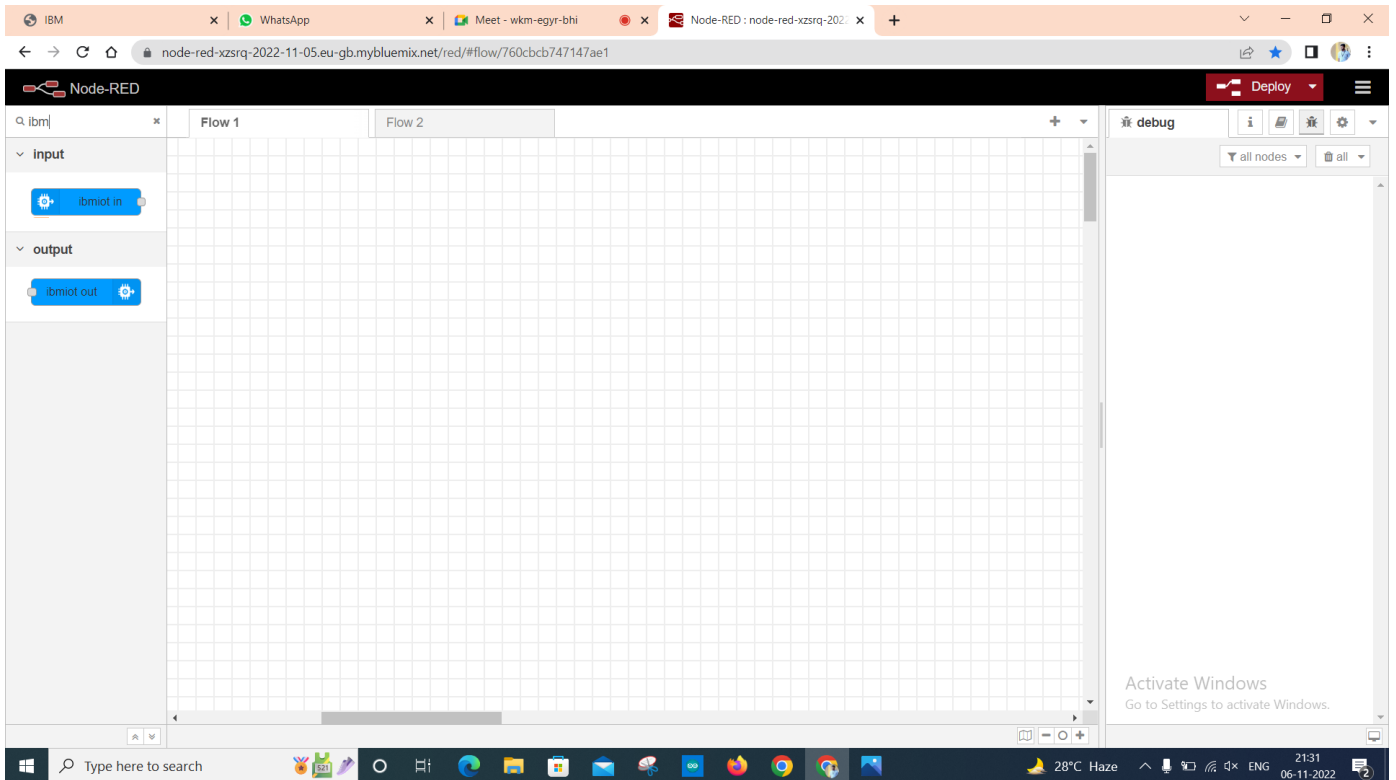
- To install IBM nodes in Node-red flow editor click on manage palette in the menu option which is on the top-right of the screen.



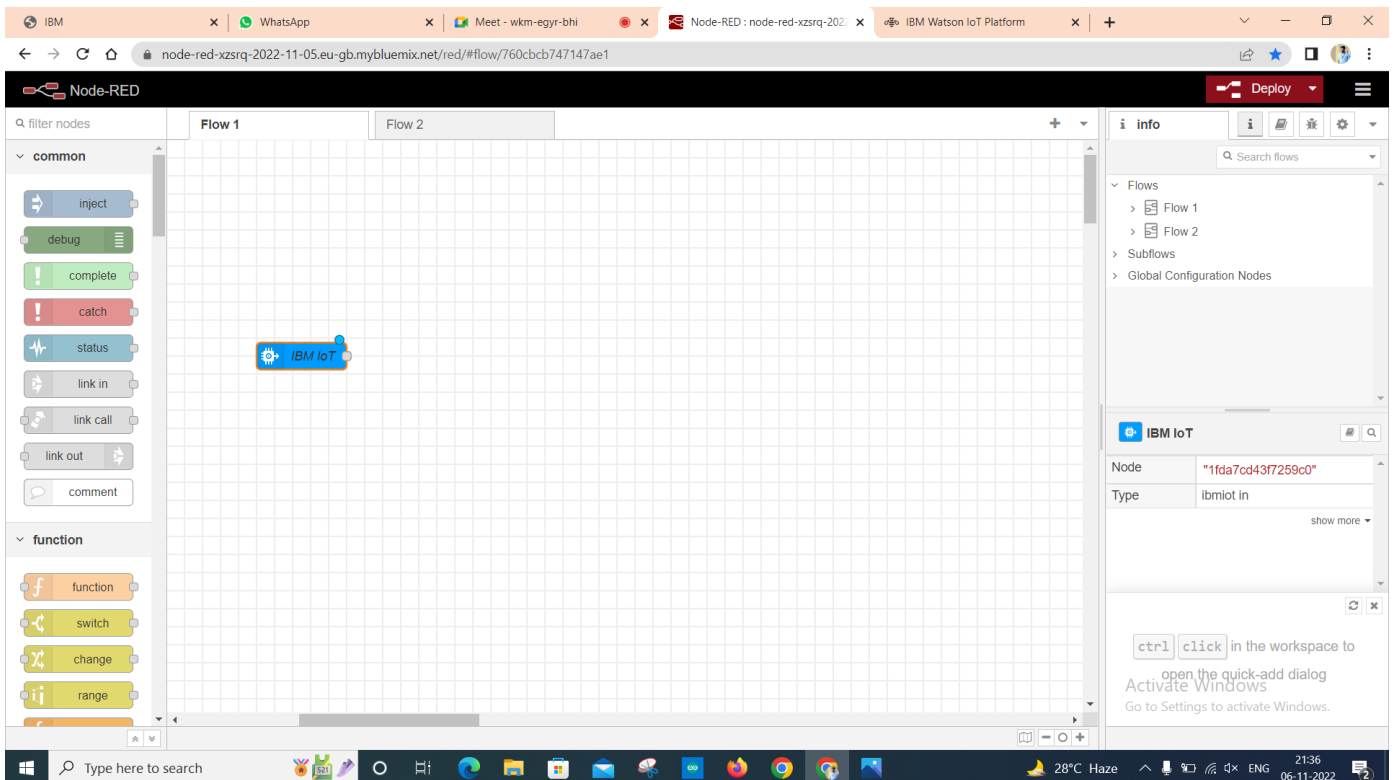
- In install section search for ibmiot and install the ibm nodes to flow editor.



➤ 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 'yf0dyy'. The 'Generate API Key' button is visible in the top right. Below it, a table lists API keys. The first key, 'a-yf0dyy-iwy9pm96o', is selected, and its details are shown in the 'API Key Information' section.

Key	Description	Role	Expires
a-yf0dyy-iwy9pm96o	API Key for the device simulator	Standard Application	-
a-yf0dyy-tbwwm8i7z2	-	Standard Application	-

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

Field	Value	Field	Value
Key	a-yf0dyy-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 'debug' console on the right displays the received data, which is a JSON object containing sensor readings.

```

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: 83, 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' (labeled 'connected') and 'msg.payload'. The left sidebar shows the 'common' and 'function' node palettes. The right sidebar shows the 'debug' tab, which contains a list of messages. A red box highlights a portion of the messages in the debug console, showing the following data:

```
msg.payload: Object
{
  Hazardous Gas: 98, Temperature:
96, Humidity: 83, Pressure: 100 }
11/6/2022, 8:29:05 PM node: b0ec530fac71d47
iot-2/type/Kumaranid/12345/ev/vent_1/fmt/json :
msg.payload: Object
{
  Hazardous Gas: 37, Temperature:
13, Humidity: 83, Pressure: 18 }
11/6/2022, 8:29:08 PM node: b0ec530fac71d47
iot-2/type/Kumaranid/12345/ev/vent_1/fmt/json :
msg.payload: Object
{
  Hazardous Gas: 18, Temperature:
59, Humidity: 20, Pressure: 60 }
11/6/2022, 8:29:11 PM node: b0ec530fac71d47
iot-2/type/Kumaranid/12345/ev/vent_1/fmt/json :
msg.payload: Object
{
  Hazardous Gas: 65, Temperature:
83, Humidity: 98, Pressure: 52 }
11/6/2022, 8:29:14 PM node: b0ec530fac71d47
iot-2/type/Kumaranid/12345/ev/vent_1/fmt/json :
msg.payload: Object
{
  Hazardous Gas: 83, Temperature:
74, Humidity: 83, Pressure: 73 }
11/6/2022, 8:29:17 PM node: b0ec530fac71d47
iot-2/type/Kumaranid/12345/ev/vent_1/fmt/json :
msg.payload: Object
{
  Hazardous Gas: 100, Temperature:
10, Humidity: 0, Pressure: 87 }
```

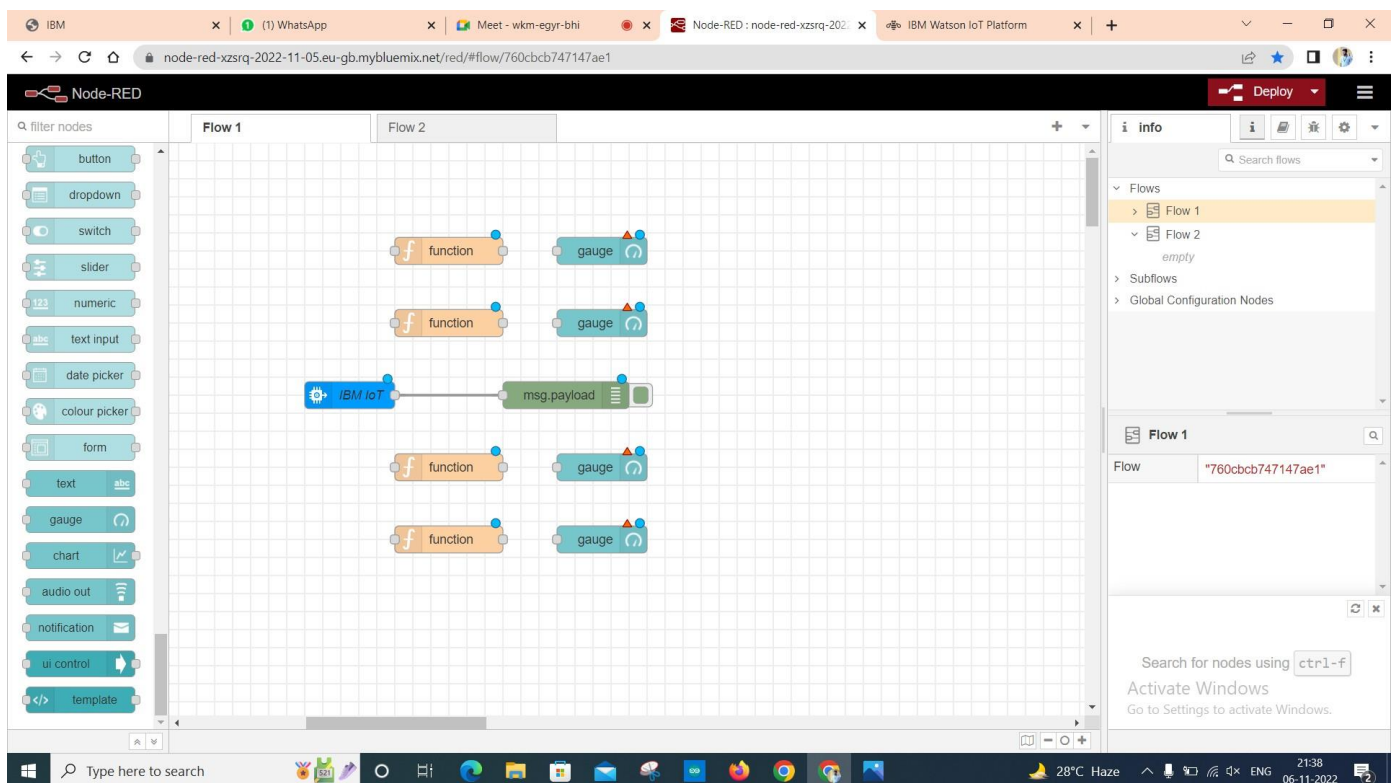
- Install the dashboard node from the manage pallet to create a UI to display temperature and humidity values in the Dashboard

The screenshot shows the Node-RED web interface with the 'manage' pallet open. The 'dashboard' node is selected in the search results. The 'User Settings' dialog is open, showing a list of dashboard-related nodes for installation. The nodes listed are:

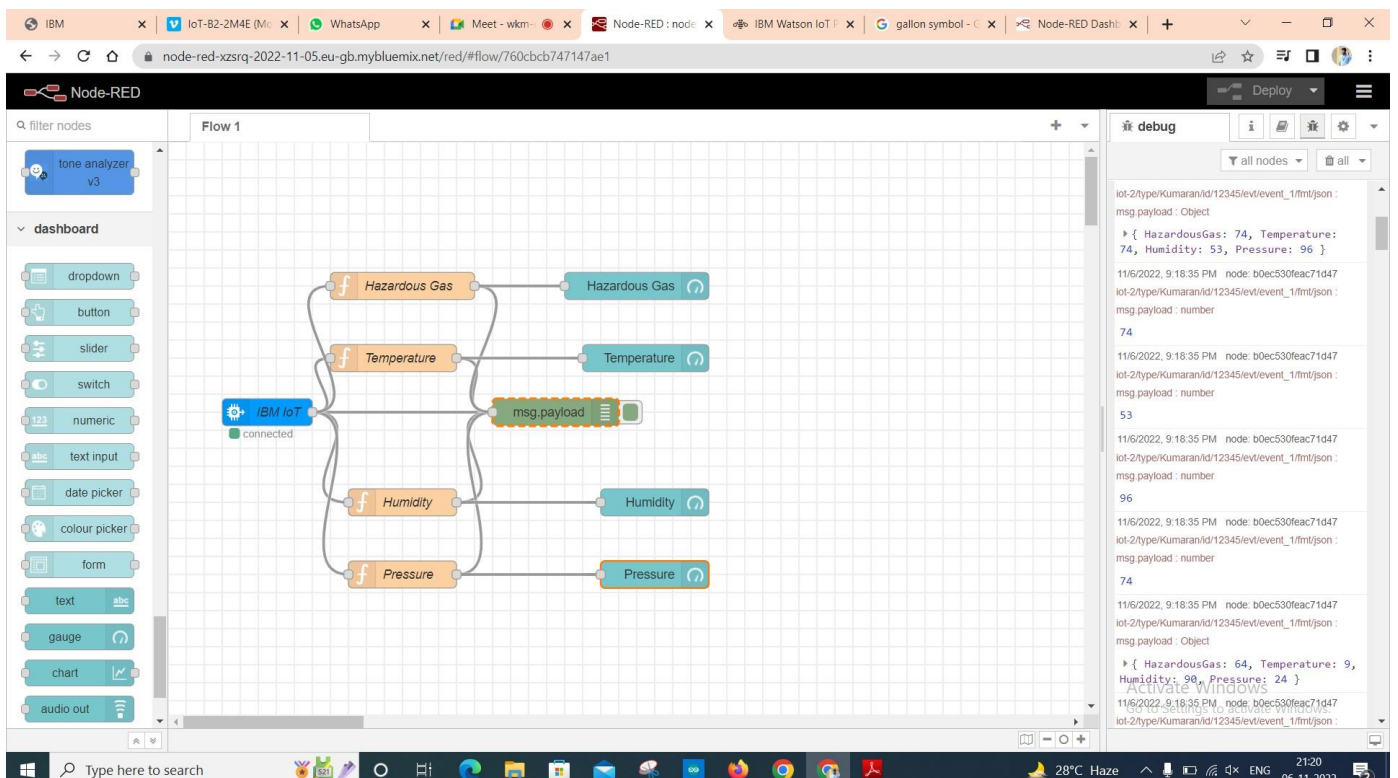
- node-red-dashboard (A set of dashboard nodes for Node-RED, 3.2.0, 2 months ago, installed)
- feezal (Web Components based Dashboard UI with WYSIWYG Editor, 0.8.1, 1 year, 11 months ago, install)
- node-red-contrib-dashboard-average-bars (Calculate and display the average values of msg payload in a bar chart, 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, 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, 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'. On the left, the 'filter nodes' sidebar is open, showing a list of nodes including 'tone analyzer v3', 'dropdown', 'button', 'slider', 'switch', 'numeric', 'text input', 'date picker', 'colour picker', 'form', 'text', 'gauge', 'chart', and 'audio out'. The central workspace contains a flow starting with an 'IBM IoT' node (labeled 'connected'). This node connects to four function nodes: 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. Each function node is connected to a corresponding gauge node on the right. A 'msg.payload' node is also connected to the function nodes. The right sidebar shows the 'debug' console with a list of messages. The bottom status bar indicates the system time as 21:20 on 06-11-2022.

- 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' set to '[Hazardous Gas] Gas Leakage', 'Size' set to 'auto', 'Type' set to 'Gauge', 'Label' set to 'Hazardous Gas', 'Value format' set to '{{value}}', 'Units' set to 'ppm', 'Range' with 'min' 0 and 'max' 100, 'Colour gradient' with a green-to-red gradient, 'Sectors' set to 0, optional, optional, 100, 'Class' set to 'Optional CSS class name(s) for widget', and 'Name' set to an empty field. The right sidebar shows the 'info' panel with a search bar and a list of flows. The bottom status bar indicates the system time as 21:40 on 06-11-2022.

➤ After editing the nodes, deploy it

The screenshot displays the Node-RED web interface in a browser. The top bar shows the Node-RED logo and a 'Deploy' button. The left sidebar contains a 'filter nodes' search bar and a 'dashboard' section with various widgets like dropdown, button, slider, switch, numeric, text input, date picker, colour picker, form, text, gauge, chart, and audio out. The main workspace shows a flow named 'Flow 1' with the following components: an 'IBM IoT' node (labeled 'connected'), five function nodes labeled 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure', a 'msg.payload' node, and five output nodes labeled 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. The flow is connected as follows: 'IBM IoT' connects to all four function nodes. Each function node connects to its corresponding output node. The 'msg.payload' node is also connected to the four function nodes. The right sidebar shows the 'debug' console with a list of messages, including JSON objects for 'HazardousGas', 'Temperature', 'Humidity', and 'Pressure'.

## RESULT:

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