

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

Date	07 November 2022
Team ID	PNT2022TMID27071
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 (Kumaran N T's Account). The main content area is titled "Resource list" and features a table with columns: Name, Group, Location, Product, Status, and Tags. A sidebar on the left lists various resource categories like Compute, Containers, Networking, Storage, AI / Machine Learning, Analytics, Blockchain, Databases, Developer tools, Logging and monitoring, and Migration. The table shows a single resource: "Node RED XZSRQ 2022-11-05" under the "Gas Leakage Monitoring / Kumaran" group, located in "London", with the product "Node.js" and status "Started". A "Create resource" button is visible in the top right corner of the resource list section. The bottom of the image shows a Windows taskbar with various application icons and system information.

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](#)

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 cost

Current and estimated cost excludes connected services.

\$0.00 \$0.00

Connections (1)

[node-red-fdeflt-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'. The address bar shows the URL 'node-red-xzsrq-2022-11-05.eu-gb.mybluemix.net'. The page has a dark red header with the text 'Node-RED on IBM Cloud'. Below this is a large red banner with the 'Node-RED' logo and the tagline 'Flow-based programming for the Internet of Things'. The main content area is light gray and contains three paragraphs of text: '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.', and 'More information about Node-RED, including documentation, can be found at nodered.org.' To the right of the text is a button that says 'Go to your Node-RED flow editor' and a link that says '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.

[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 address bar shows the URL '127.0.0.1:1880/#flow/b47948623bf1c79f'. The interface is divided into several sections: a left sidebar with a search bar and two categories of nodes ('common' and 'function'), a central workspace with a grid and a flow diagram, and a right sidebar with an 'info' panel and a 'Flow 1' panel. The 'common' category includes nodes like 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', and 'comment'. The 'function' category includes nodes like 'function', 'switch', 'change', and 'range'. The 'info' panel shows a search bar and a list of flows, with 'Flow 1' selected. The 'Flow 1' panel shows the flow ID 'b47948623bf1c79f' and a message box that says 'Move the selected nodes using the ↑ ↓ and ← → keys. Hold ⇧ to nudge'. At the bottom, there is a Windows taskbar with various icons and a system tray showing the date and time.

Node-RED

filter nodes

Flow 1

common

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

function

- function
- switch
- change
- range

info

Search flows

Flows

- Flow 1
- Subflows
- Global Configuration Nodes

Flow 1

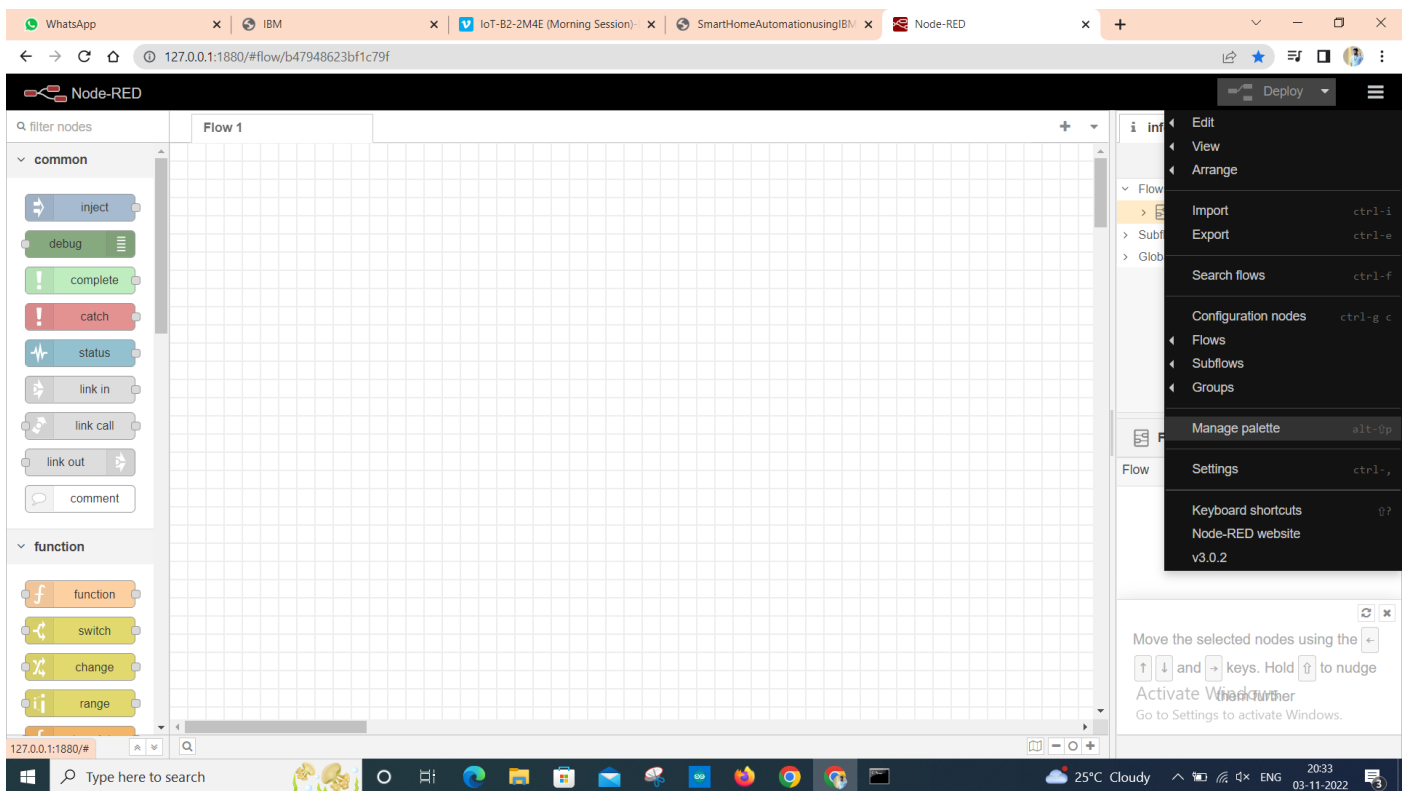
Flow

"b47948623bf1c79f"

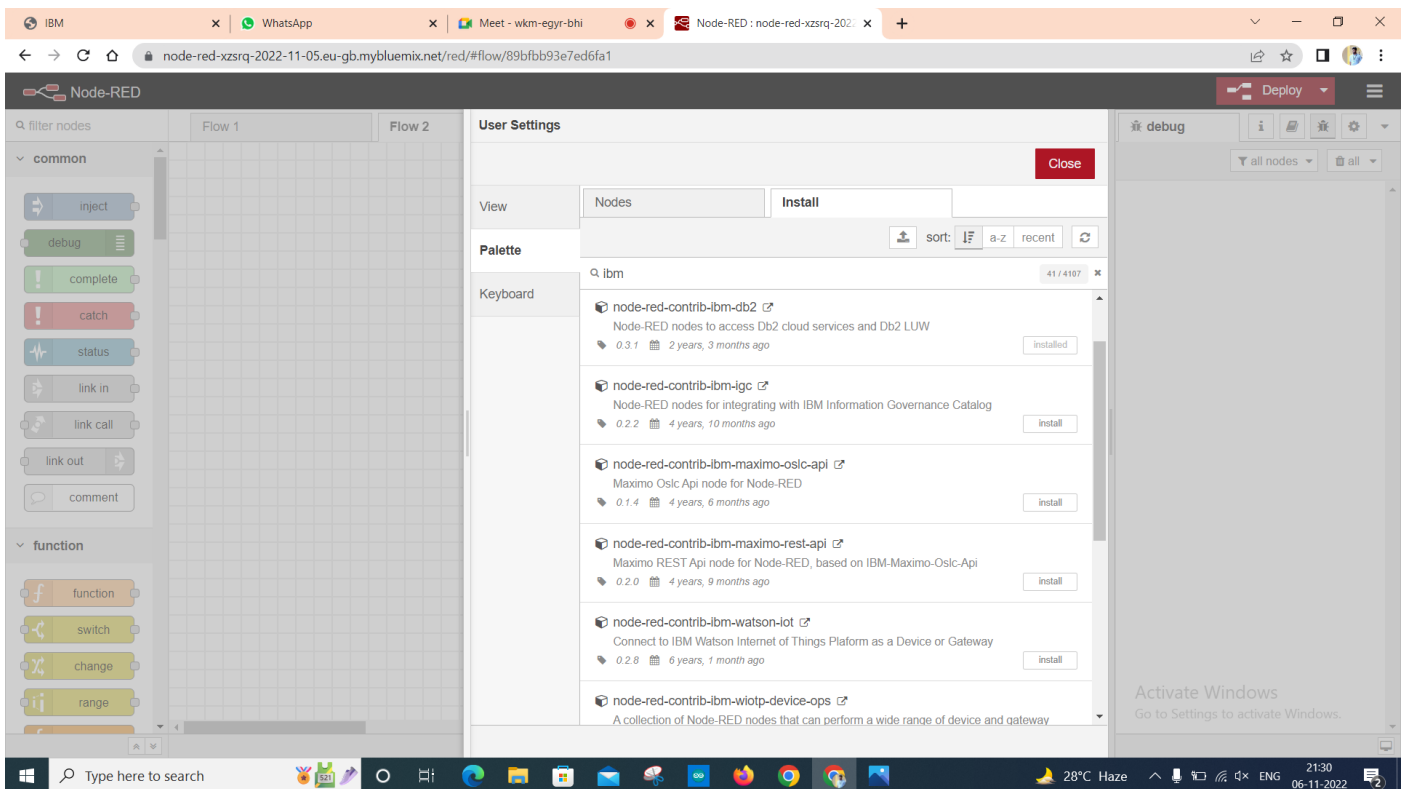
Move the selected nodes using the ↑ ↓ and ← → keys. Hold ⇧ to nudge

Activate Windows
Go to Settings to activate Windows.

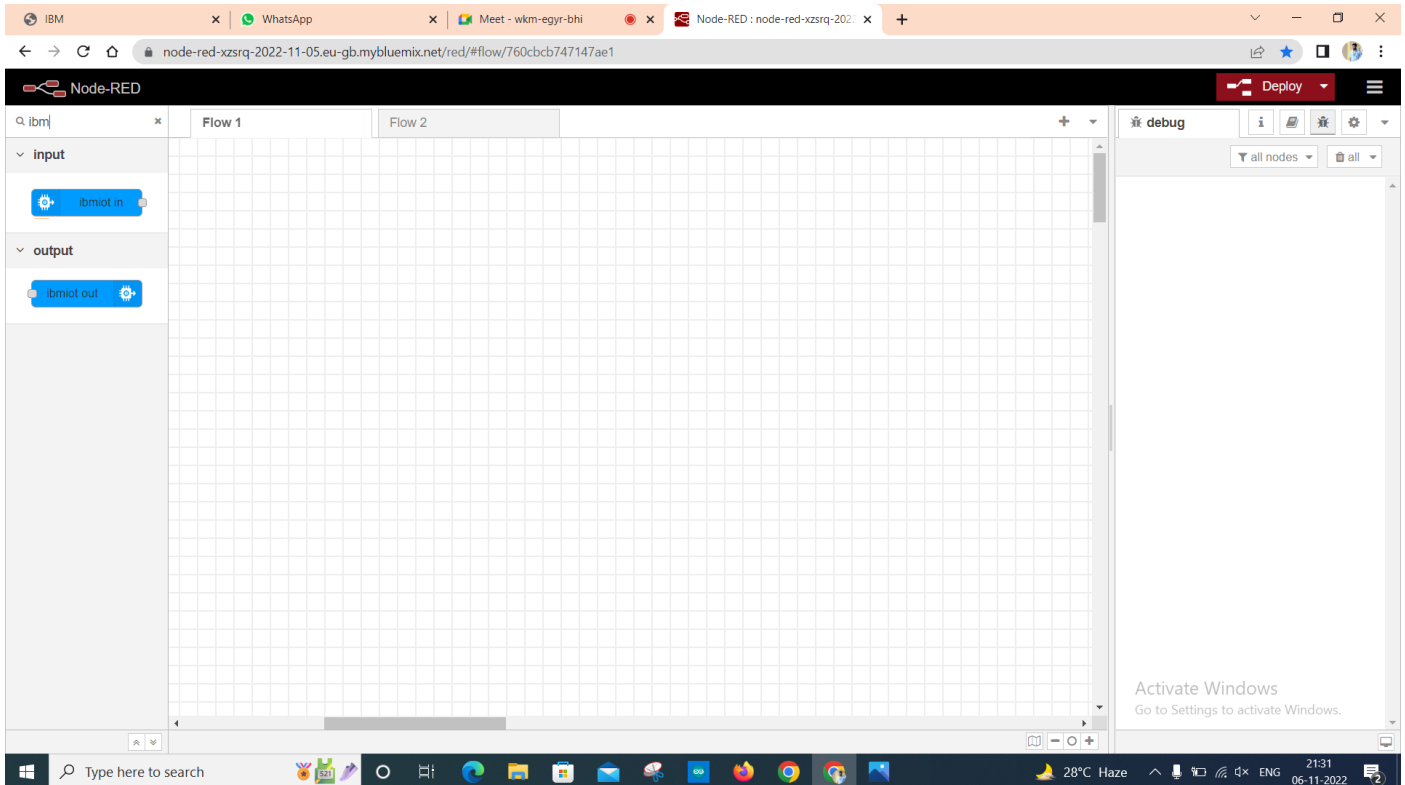
- To install IBM nodes in Node-red flow editor click on manage palette in the menu option which is on the top-right of the screen.



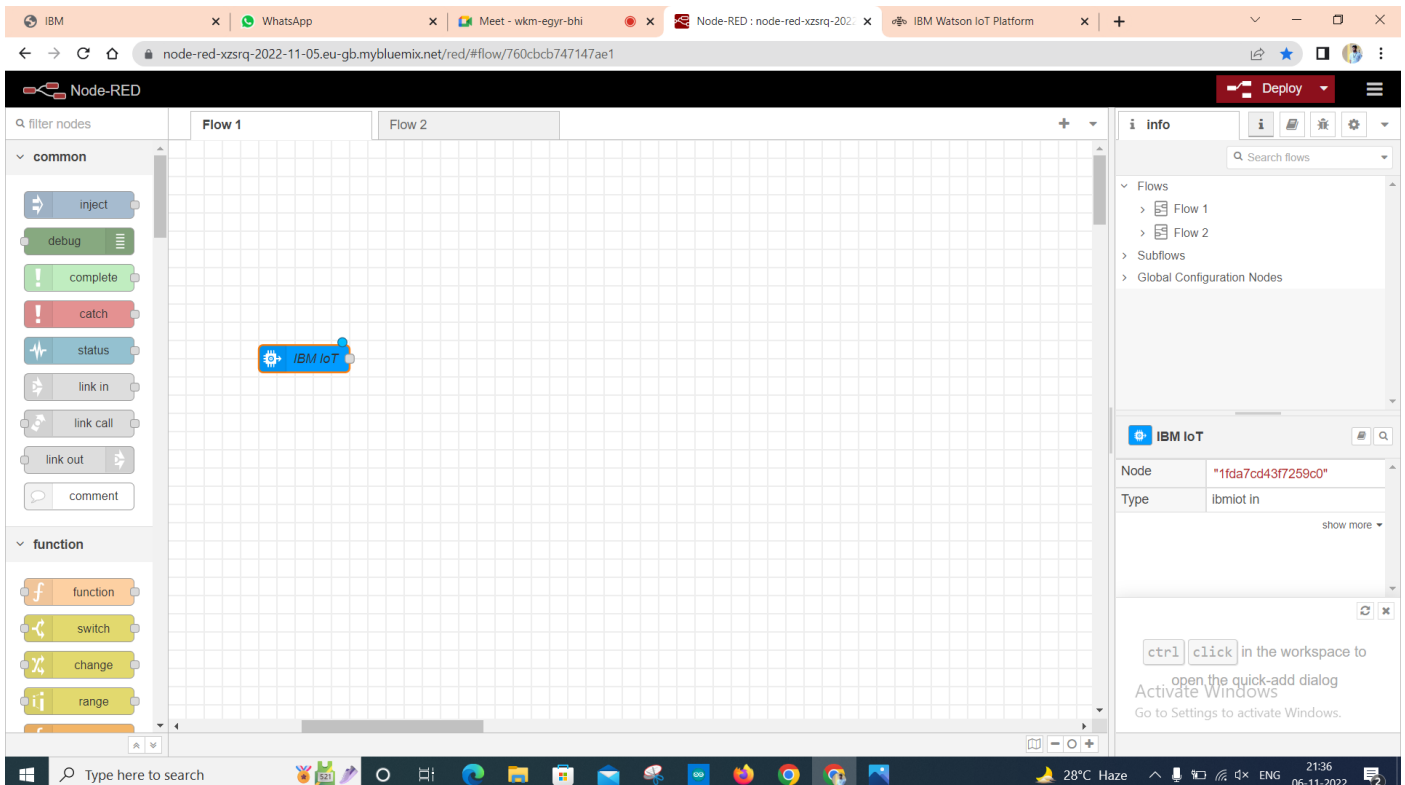
- In install section search for ibmiot and install the ibm nodes to flow editor.



➤ 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 corner. Below the header, there is a table listing API keys. The first key is 'a-yf0ddy-iwy9pm96o' with the description 'API Key for the device simulator' and role 'Standard Application'. The 'API Key Information' tab is selected, showing details for the key 'a-yf0ddy-iwy9pm96o':

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

Below the table, there is a button 'Generate API Key' and a status indicator '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. The 'Flow 1' tab is active, displaying a flow with two nodes: 'IBM IoT' and 'msg.payload'. The 'IBM IoT' node is connected. The 'Deploy' button is visible in the top right corner. The 'debug' console on the right shows the following log entries:

```

msg payload : Object
{
  Hazardous Gas: 98, Temperature: 96, Humidity: 83, Pressure: 100
}
11/6/2022, 8:29:05 PM node: b0ec530feac71d47
iot-2/type/Kumaranid/12345/ev/levent_1/fmt/json :
msg payload : Object
{
  Hazardous Gas: 37, Temperature: 13, Humidity: 83, Pressure: 18
}
11/6/2022, 8:29:08 PM node: b0ec530feac71d47
iot-2/type/Kumaranid/12345/ev/levent_1/fmt/json :
msg payload : Object
{
  Hazardous Gas: 18, Temperature: 59, Humidity: 20, Pressure: 60
}
11/6/2022, 8:29:11 PM node: b0ec530feac71d47
iot-2/type/Kumaranid/12345/ev/levent_1/fmt/json :
msg payload : Object
{
  Hazardous Gas: 65, Temperature: 83, Humidity: 98, Pressure: 52
}
11/6/2022, 8:29:14 PM node: b0ec530feac71d47
iot-2/type/Kumaranid/12345/ev/levent_1/fmt/json :
msg payload : Object
{
  Hazardous Gas: 83, Temperature: 74, Humidity: 83, Pressure: 73
}
11/6/2022, 8:29:17 PM node: b0ec530feac71d47
iot-2/type/Kumaranid/12345/ev/levent_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 is displaying a list of messages. A red box highlights the messages in the debug tab, 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
msg.payload : Object
  { Hazardous Gas: 37, Temperature:
    13, Humidity: 83, Pressure: 18 }
11/6/2022 8:29:08 PM node: b0ec530fac71d47
msg.payload : Object
  { Hazardous Gas: 18, Temperature:
    59, Humidity: 20, Pressure: 60 }
11/6/2022 8:29:11 PM node: b0ec530fac71d47
msg.payload : Object
  { Hazardous Gas: 65, Temperature:
    83, Humidity: 98, Pressure: 32 }
11/6/2022 8:29:14 PM node: b0ec530fac71d47
msg.payload : Object
  { Hazardous Gas: 83, Temperature:
    74, Humidity: 83, Pressure: 73 }
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 search bar contains the text 'dashboard'. The search results show the following nodes:

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

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

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 a 'msg.payload' node (green). This is followed by two 'function' nodes (orange) and two 'gauge' nodes (blue). The left sidebar contains a 'filter nodes' search bar and a list of available nodes including 'button', 'dropdown', 'switch', 'slider', 'numeric', 'text input', 'date picker', 'colour picker', 'form', 'text', 'gauge', 'chart', 'audio out', 'notification', 'ui control', and 'template'. The right sidebar shows the 'info' panel with a search bar and a list of flows, including 'Flow 1' and 'Flow 2'. The bottom status bar shows the system clock as 21:38 on 06-11-2022.

- 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

The screenshot shows the Node-RED web interface after the flow has been updated. The main workspace displays a flow named 'Flow 1'. It starts with an 'IBM IoT' node (blue) connected to a 'msg.payload' node (green). This is followed by four 'function' nodes (orange) labeled 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. Each function node is connected to a corresponding 'gauge' node (blue) labeled 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. The left sidebar contains a 'filter nodes' search bar and a list of available nodes including 'tone analyzer v3', 'dropdown', 'button', 'slider', 'switch', 'numeric', 'text input', 'date picker', 'colour picker', 'form', 'text', 'gauge', 'chart', 'audio out', and 'template'. The right sidebar shows the 'debug' panel with a search bar and a list of debug messages, including 'HazardousGas: 74, Temperature: 96' and 'HazardousGas: 64, Temperature: 9, Humidity: 90, Pressure: 24'. The bottom status bar shows the system clock as 21:20 on 06-11-2022.

- 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) which is connected to four function nodes (orange) labeled 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. Each function node is connected to a corresponding gauge node (blue) on the right. The left sidebar shows the 'dashboard' category with various nodes like dropdown, button, slider, switch, numeric, text input, date picker, colour picker, form, text, gauge, chart, and audio out. The right sidebar shows the 'debug' console with a log of messages from the IoT node, including timestamps and payloads like 'msg.payload : number' and 'msg.payload : Object'.

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

This screenshot shows the same Node-RED interface, but with the 'Edit gauge node' dialog box open for the 'Hazardous Gas' gauge. The dialog has a 'Delete' button, 'Cancel' and 'Done' buttons, and a 'Properties' section. 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 gradient from green to red), Sectors (0, optional, optional, 100), Class ('Optional CSS class name(s) for widget'), and Name (an empty field). The background shows the same flow as the previous screenshot, but the 'Hazardous Gas' gauge node is highlighted. The right sidebar shows the 'info' panel with a search bar and a list of flows, including 'Flow 1' and 'Flow 2'.

➤ After editing the nodes, deploy it

The screenshot displays the Node-RED web interface in a browser. The top navigation bar includes a 'Deploy' button. The left sidebar shows a 'filter nodes' search bar and a 'dashboard' section with various widget options like dropdown, button, slider, switch, numeric, text input, date picker, colour picker, form, text, gauge, chart, and audio out. The main workspace, titled 'Flow 1', contains a flow starting with an 'IBM IoT' node (labeled 'connected'). This node feeds into four function nodes: 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. Each function node is connected to a corresponding output node: 'Hazardous Gas' to 'Hazardous Gas', 'Temperature' to 'Temperature', 'Humidity' to 'Humidity', and 'Pressure' to 'Pressure'. A 'msg.payload' node is also connected to the function nodes. The right sidebar shows a '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.