

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

Date	07 November 2022
Team ID	PNT2022TMID28936
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 links for Catalog, Manage, and the user account (Kumaran N T's Account). The main content area is titled "Resource list" and features a table with columns for 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 group "Gas Leakage Monitoring / Kumaran", located in "London", using "Node.js", with a status of "Started". A "Create resource" button is visible in the top right corner of the resource list section. The Windows taskbar at the bottom shows the system clock as 21:48 on 06-11-2022.

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

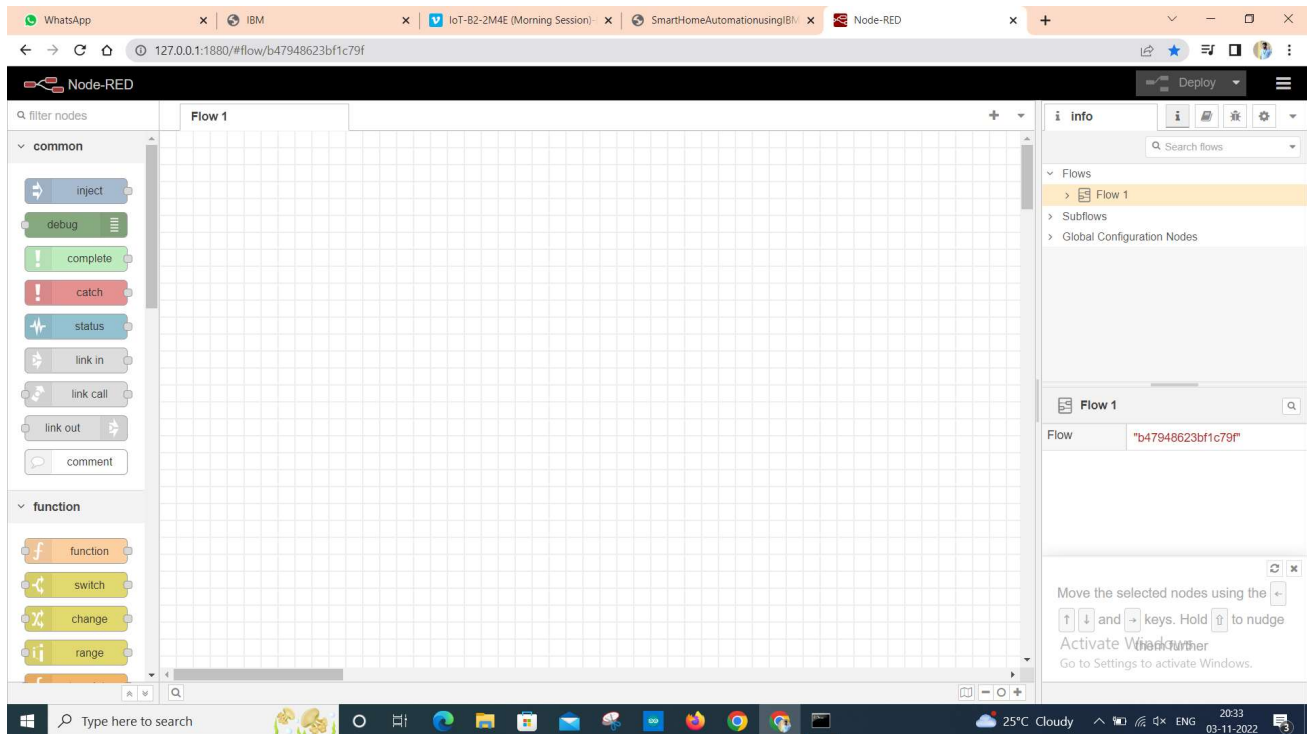
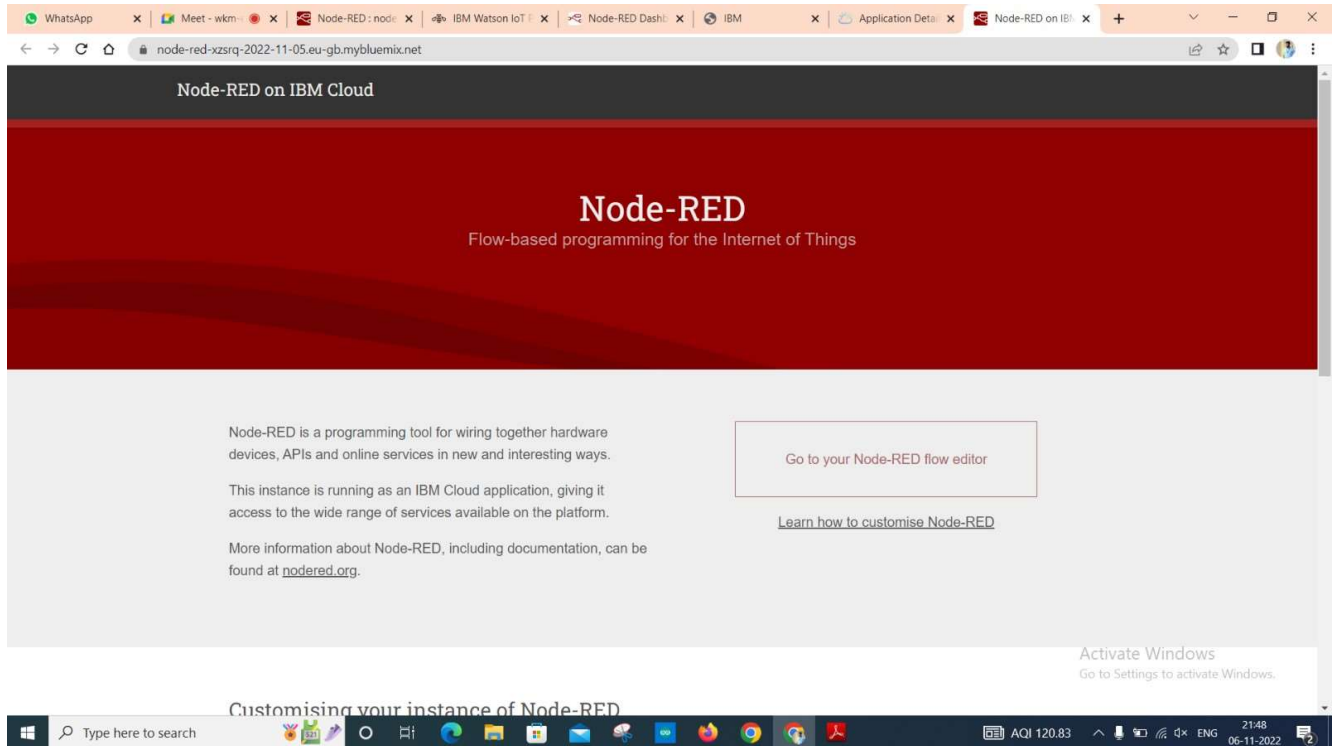
- A new window appears where we need to NODE-RED SELDZ app created before.

The screenshot shows the IBM Cloud console interface. At the top, there's a navigation bar with the IBM Cloud logo and a search bar. Below it, the 'Resource list' page is displayed. The page has a sidebar on the left with various resource categories like Compute, Networking, Storage, etc. The main area shows a table of resources. One resource is highlighted: 'Node RED XZSRQ 2022-11-05'. The table columns are Name, Group, Location, Product, Status, and Tags. The resource is in the 'Compute' group, located in 'London', with the product 'Node.js' and status 'Started'. A 'Create resource' button is visible in the top right corner.

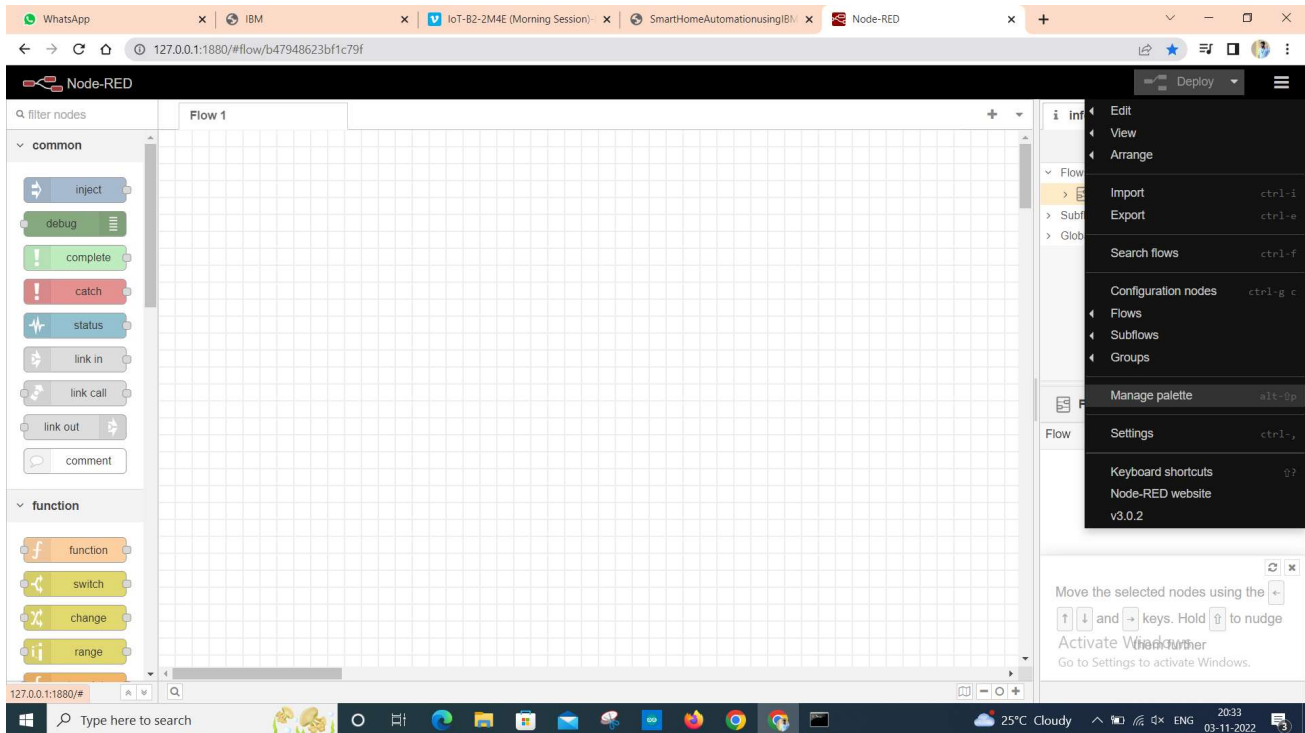
- Click on Visit App URL in Node RED SELDZ service dashboard.

The screenshot shows the 'Node RED XZSRQ 2022-11-05' service dashboard. The dashboard has a sidebar on the left with navigation links like Overview, Runtime, Connections, Logs, API Management, and Autoscaling. The main area displays various metrics and status information. At the top, it says 'Node RED XZSRQ 2022-11-05' with a 'Running' status and a 'Visit App URL' link. Below this, there's a section for 'Instances' showing 'Health' at '100%' and '1/1 instance(s) are running'. There's also a 'Runtime' section showing 'Node.js' with a '256' MB allocation. A 'Connections (1)' section shows a single connection. The dashboard also displays 'Runtime cost' as '\$0.00' and 'Estimated total for billing period' as '\$0.00'. An 'Activate Windows' watermark is visible in the bottom right corner.

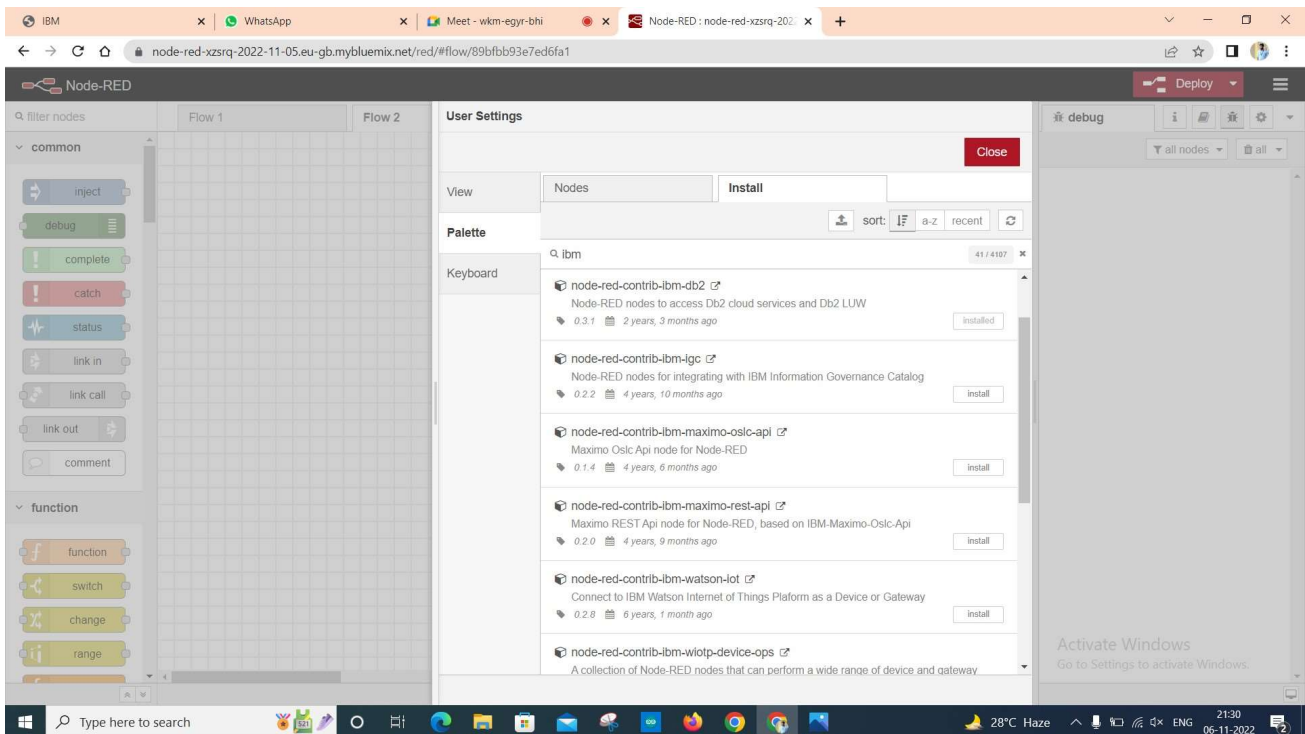
- Click on your Node-RED flow editor where you will be redirected to the Node-RED flow editor.



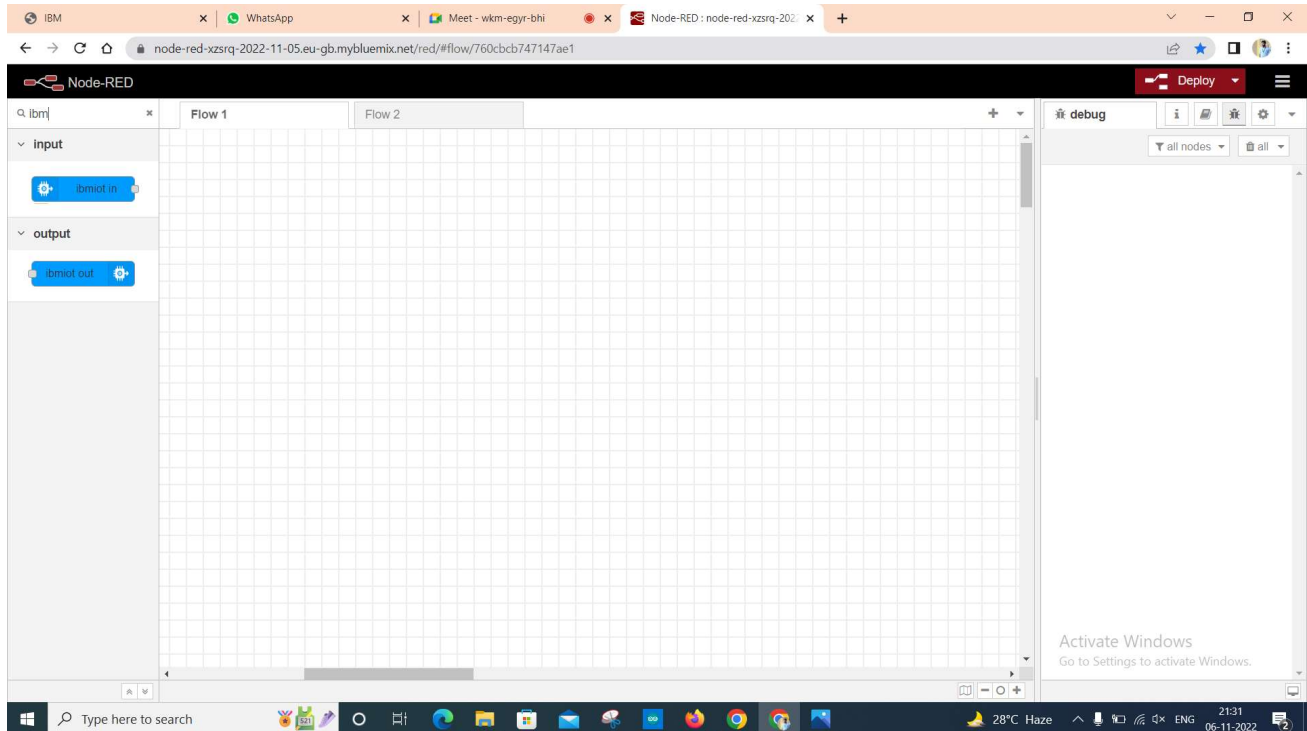
- To install IBM nodes in Node-red flow editor click on manage palette in the menu option which is on the top-right of the screen.



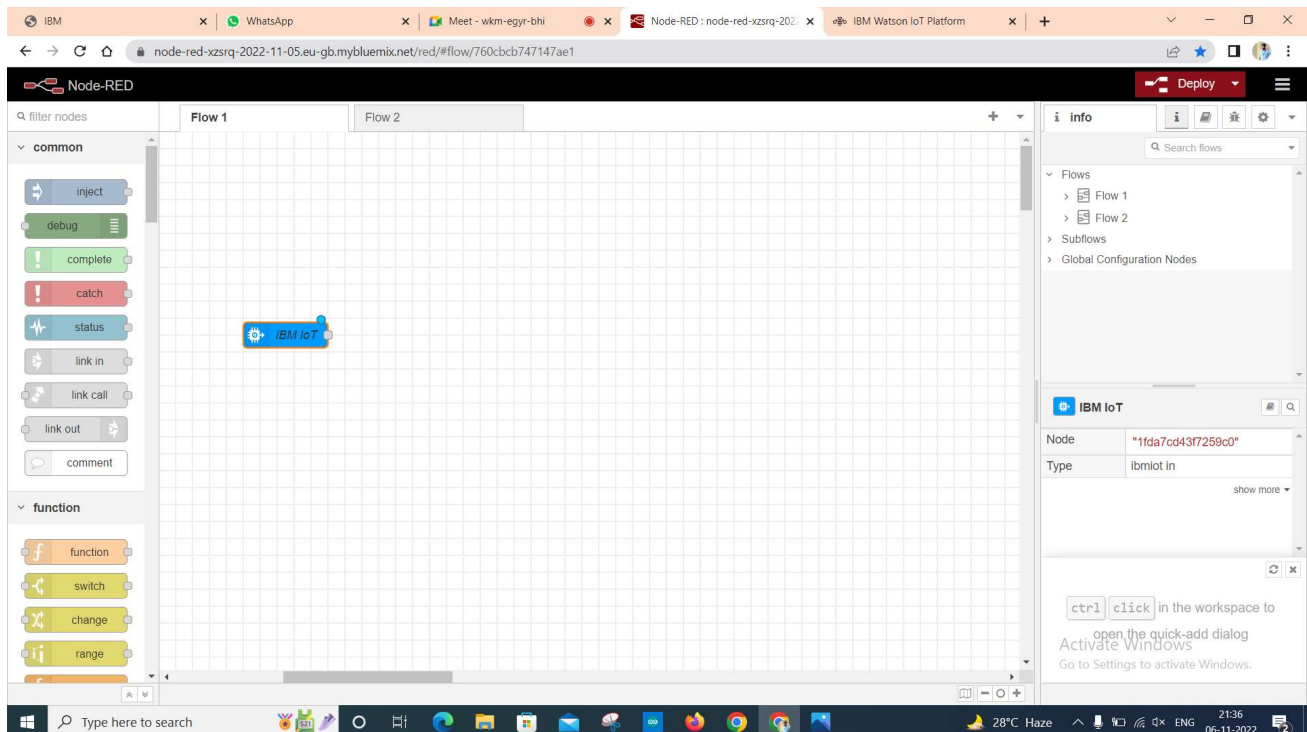
- In install section search for ibmiot and install the ibm nodes to flow editor.



- 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

The screenshot displays the 'Edit ibmiot in node' configuration window. The 'Properties' section is active, showing various settings for the node:

- 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

A yellow tooltip at the bottom of the properties section reads: "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".

At the bottom left, there is a checkbox labeled 'Enabled' which is currently unchecked.

On the right side of the interface, the 'info' panel shows the node details:

- Node:** "c9195d4cbebc02e8"
- Type:** ibmiot in

Below the node details, there is a message: "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 top navigation bar includes the IBM logo and the user's profile. The main content area is titled 'IBM Cloud Apps' and features a '+ Generate API Key' button. Below this, there is a table listing API keys. The first key is 'a-yf0ddy-iwy9pm96o' with the description 'API Key for the device simulator' and a role of 'Standard Application'. The second key is 'a-yf0ddy-tbwwm8i7z2' with a role of 'Standard Application'. The table has columns for 'Key', 'Description', 'Role', and 'Expires'. A '1 Simulation running' notification is visible at the bottom right of the dashboard.

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

- 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 main workspace displays a flow named 'Flow 1' with two nodes: 'IBM IoT' and 'msg.payload'. The 'IBM IoT' node is marked as 'connected'. The left sidebar shows the 'common' and 'function' node categories. The right sidebar shows the 'debug' console, which displays a series of messages from the 'IBM IoT' node, including 'Hazardous Gas: 98, Temperature: 96, Humidity: 83, Pressure: 100' and 'Hazardous Gas: 100, Temperature: 10, Humidity: 0, Pressure: 87'.

```

msg.payload: Object
  { Hazardous Gas: 98, Temperature: 96, Humidity: 83, Pressure: 100 }
11/6/2022, 8:29:05 PM node:b0ec530feac71d47
iot-2/type/Kumaran/ld12345/ev/evEvent_1/mj/mjjson :
msg.payload: Object
  { Hazardous Gas: 37, Temperature: 13, Humidity: 83, Pressure: 18 }
11/6/2022, 8:29:06 PM node:b0ec530feac71d47
iot-2/type/Kumaran/ld12345/ev/evEvent_1/mj/mjjson :
msg.payload: Object
  { Hazardous Gas: 18, Temperature: 59, Humidity: 20, Pressure: 60 }
11/6/2022, 8:29:11 PM node:b0ec530feac71d47
iot-2/type/Kumaran/ld12345/ev/evEvent_1/mj/mjjson :
msg.payload: Object
  { Hazardous Gas: 65, Temperature: 83, Humidity: 98, Pressure: 52 }
11/6/2022, 8:29:14 PM node:b0ec530feac71d47
iot-2/type/Kumaran/ld12345/ev/evEvent_1/mj/mjjson :
msg.payload: Object
  { Hazardous Gas: 83, Temperature: 74, Humidity: 83, Pressure: 73 }
11/6/2022, 8:29:17 PM node:b0ec530feac71d47
iot-2/type/Kumaran/ld12345/ev/evEvent_1/mj/mjjson :
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 has two nodes: 'IBM IoT' and 'msg.payload'. The 'debug' tab on the right is active, showing a list of messages. A red box highlights a portion of the messages, which are JSON objects containing sensor data. The messages are as follows:

```
11/6/2022, 8:29:05 PM node: b0ec530feac71d47  
iot-2/type/Kumaran/12345/ev/Event_1/fmt/json :  
msg.payload : Object  
{  
  Hazardous Gas: 98, Temperature: 90,  
  Humidity: 83, Pressure: 100  
}  
11/6/2022, 8:29:08 PM node: b0ec530feac71d47  
iot-2/type/Kumaran/12345/ev/Event_1/fmt/json :  
msg.payload : Object  
{  
  Hazardous Gas: 37, Temperature: 13,  
  Humidity: 83, Pressure: 18  
}  
11/6/2022, 8:29:11 PM node: b0ec530feac71d47  
iot-2/type/Kumaran/12345/ev/Event_1/fmt/json :  
msg.payload : Object  
{  
  Hazardous Gas: 18, Temperature: 59,  
  Humidity: 20, Pressure: 60  
}  
11/6/2022, 8:29:14 PM node: b0ec530feac71d47  
iot-2/type/Kumaran/12345/ev/Event_1/fmt/json :  
msg.payload : Object  
{  
  Hazardous Gas: 65, Temperature: 85,  
  Humidity: 98, Pressure: 52  
}  
11/6/2022, 8:29:17 PM node: b0ec530feac71d47  
iot-2/type/Kumaran/12345/ev/Event_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/Kumaran/12345/ev/Event_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 'User Settings' dialog box open. The 'Nodes' tab is selected, and the 'Install' button is visible. A search bar contains the text 'dashboard'. The list of nodes includes:

- 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 0.0.8, 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

The screenshot shows the Node-RED web interface in a browser. The left sidebar contains a 'filter nodes' search bar and a list of 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 main workspace shows 'Flow 1' with the following nodes: an 'IBM IoT' node (blue), a 'msg.payload' node (green), and four 'function' nodes (orange) connected in a sequence. Each 'function' node is connected to a 'gauge' node (blue). 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 modifications. The left sidebar now includes a 'dashboard' section with nodes like dropdown, button, slider, switch, numeric, text input, date picker, colour picker, form, text, gauge, chart, and audio out. The main workspace shows 'Flow 1' with the following nodes: an 'IBM IoT' node (blue), a 'msg.payload' node (green), and four 'function' nodes (orange) labeled 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. Each function node is connected to a corresponding 'gauge' node (blue) with the same label. The right sidebar shows the 'debug' panel with a search bar and a list of debug messages. 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 (connected). This node branches into four parallel function nodes: 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. Each function node is connected to a corresponding 'msg.payload' node. These four 'msg.payload' nodes are then connected to four 'gauge' nodes, each labeled with its respective sensor name: 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. 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 log of messages, including a JSON object:

```
{ HazardousGas: 74, Temperature: 74, Humidity: 53, Pressure: 96 }
```

. The bottom status bar shows the system clock 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' section. The 'Properties' section includes: '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' at 0 and 'max' at 100, 'Colour gradient' with a green-to-red gradient, 'Sectors' set to 0, 100, and 'optional' for intermediate values, '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, including 'Flow 1' and 'Flow 2'. The bottom status bar shows the system clock as 21:40 on 06-11-2022.

➤ After editing the nodes, deploy it

The screenshot displays the Node-RED web application 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 widgets 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 diagram. It starts with an 'IBM IoT' node (labeled 'connected') that branches into four parallel processing nodes: 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. Each of these nodes is connected to a corresponding output node on the right: 'Hazardous Gas', 'Temperature', 'Humidity', and 'Pressure'. A 'msg.payload' node is also present in the flow. The right sidebar shows a 'debug' console with a list of messages. The messages are JSON objects containing sensor data, such as:

```
{ HazardousGas: 74, Temperature: 74, Humidity: 53, Pressure: 96 }
```

 and

```
{ HazardousGas: 64, Temperature: 9, Humidity: 90, Pressure: 24 }
```

. The bottom of the image shows the Windows taskbar with the system clock indicating 21:20 on 06-11-2022.

RESULT:

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