

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

Date	07 November 2022
Team ID	PNT2022TMID53803
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 browser's address bar shows the URL `cloud.ibm.com/resources`. The dashboard header includes the IBM Cloud logo, a search icon, and navigation links for Catalog and Manage. A sidebar on the left contains various service icons. The main content area is titled "Resource list" and features a "Create resource" button. Below the title is a table with columns: Name, Group, Location, Product, Status, and Tags. The table includes filter input fields for each column. A list of resource categories is shown, each with a dropdown arrow and a count in parentheses: Compute (1), Containers (0), Networking (0), Storage (0), AI / Machine Learning (0), Analytics (0), Blockchain (0), and Databases (2). A blue chat icon is visible in the bottom right corner of the dashboard area.

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

The screenshot shows the IBM Cloud console interface. The top navigation bar includes the IBM Cloud logo, a search icon, and links for Catalog, Manage, and a user profile. The main content area is titled "Resource list" and features a "Create resource" button. Below the title is a table with columns: Name, Group, Location, Product, Status, and Tags. The table contains one resource: "Node RED ZUOID 2022-11-07" under the group "Nandha kumar / dev", located in "London", using the "Node.js" product, with a status of "Started". The left sidebar shows various service categories like Compute, Networking, Storage, AI / Machine Learning, Analytics, and Blockchain. A tooltip is visible over the resource name, showing "Node RED ZUOID 2022-11-07".

Name	Group	Location	Product	Status	Tags
Node RED ZUOID 2022-11-07	Nandha kumar / dev	London	Node.js	Started	—

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

O

The screenshot shows the IBM Cloud console interface for a Node RED application. The browser's address bar displays the URL: `cloud.ibm.com/apps/38cb7d6f-5ee5-492e-8c55-ef40975cdd5d?ace_config=%7B%3A%2C%3A%3A...`. The console header includes the IBM Cloud logo, a search bar, and navigation links for Catalog, Manage, and a user profile icon. The main content area is titled "Node RED ZUOID 2022-11-07" and indicates the application is "Running". A sidebar on the left lists navigation options: Getting started, Overview (selected), Runtime, Connections, Logs, API Management, and Autoscaling. A top notification bar states: "IBM Cloud Foundry Public is being deprecated. Please see [full details](#)." The "Overview" section displays the following information:

- Health:** 100% (1/1 instance(s) are running)
- Instances:** A control showing 1 instance with minus and plus buttons.
- Runtime:** A donut chart for "Node.js" showing a "Total MB allocation" of 256. Below the chart, it indicates "1.75 GB still available".
- MB memory per instance:** A slider set between 0 and 2048, with a selected value of 256.

The URL at the bottom of the console is `https://node-red-zuoid-2022-11-07.eu-gb.mybluemix.net`.

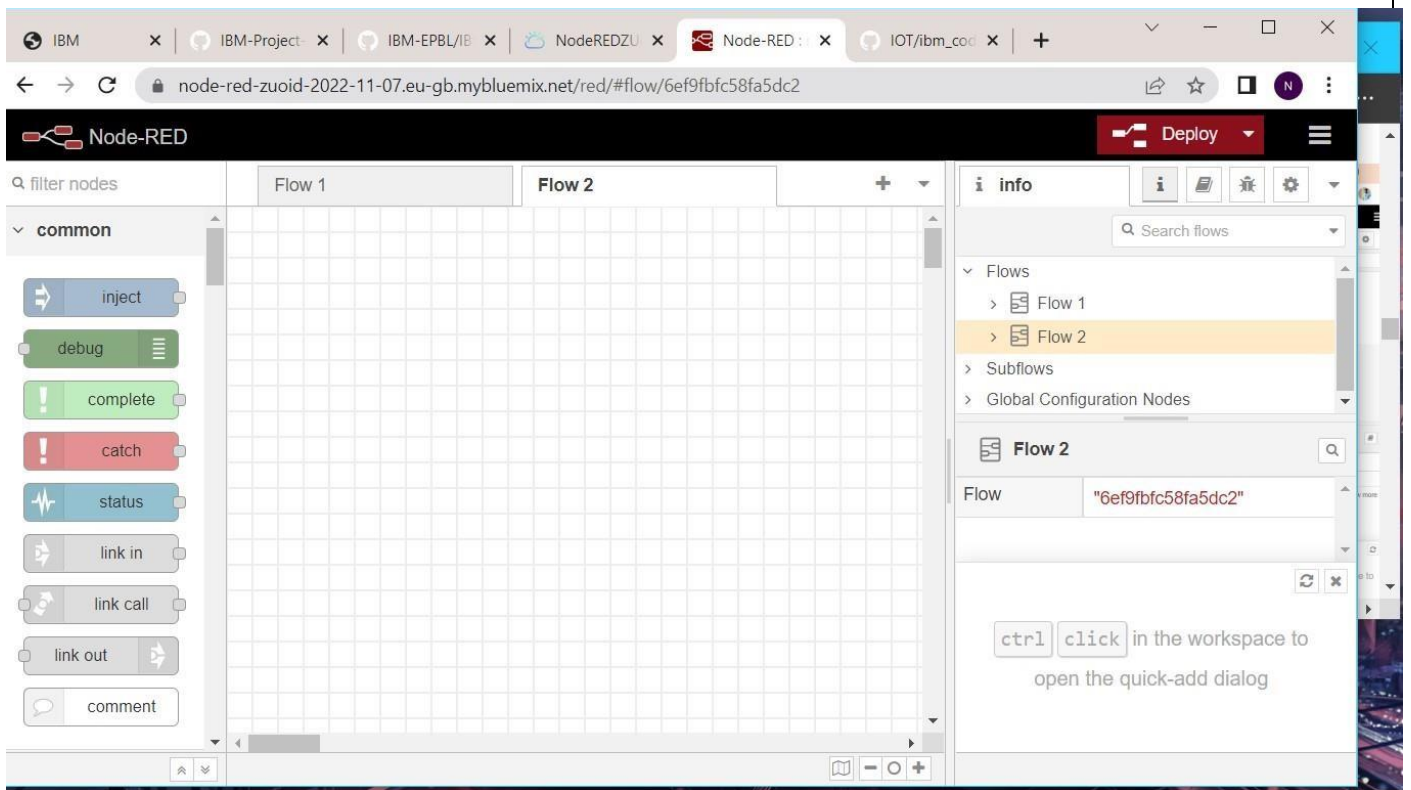
Click on your Node-RED flow editor where you will be redirected to the Node-RED flow editor.

The screenshot shows the landing page of the Node-RED flow editor. The browser's address bar displays the URL: `node-red-zuoid-2022-11-07.eu-gb.mybluemix.net`. The page has a dark red header with the text "Node-RED on IBM Cloud". The main content area features a large red banner with the text "Node-RED" and "Flow-based programming for the Internet of Things". Below the banner, there is a light gray section with the following text:

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

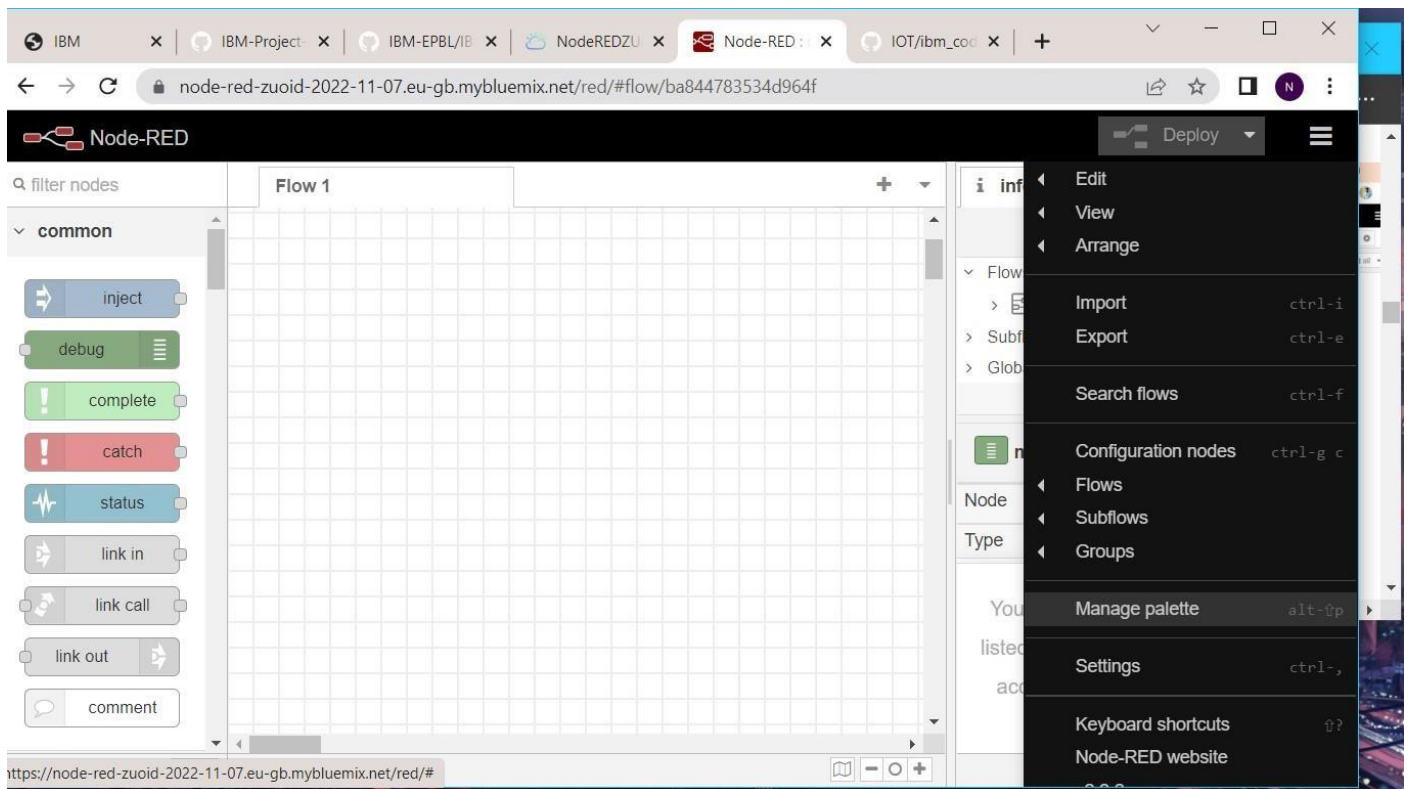
This instance is running as an IBM Cloud application, giving it access to the wide range of services available on the platform.

At the bottom right, there is a red button labeled "Go to your Node-RED flow editor" and a link labeled "Learn how to customise Node-RED". The URL at the bottom of the page is `https://node-red-zuoid-2022-11-07.eu-gb.mybluemix.net/red/`.

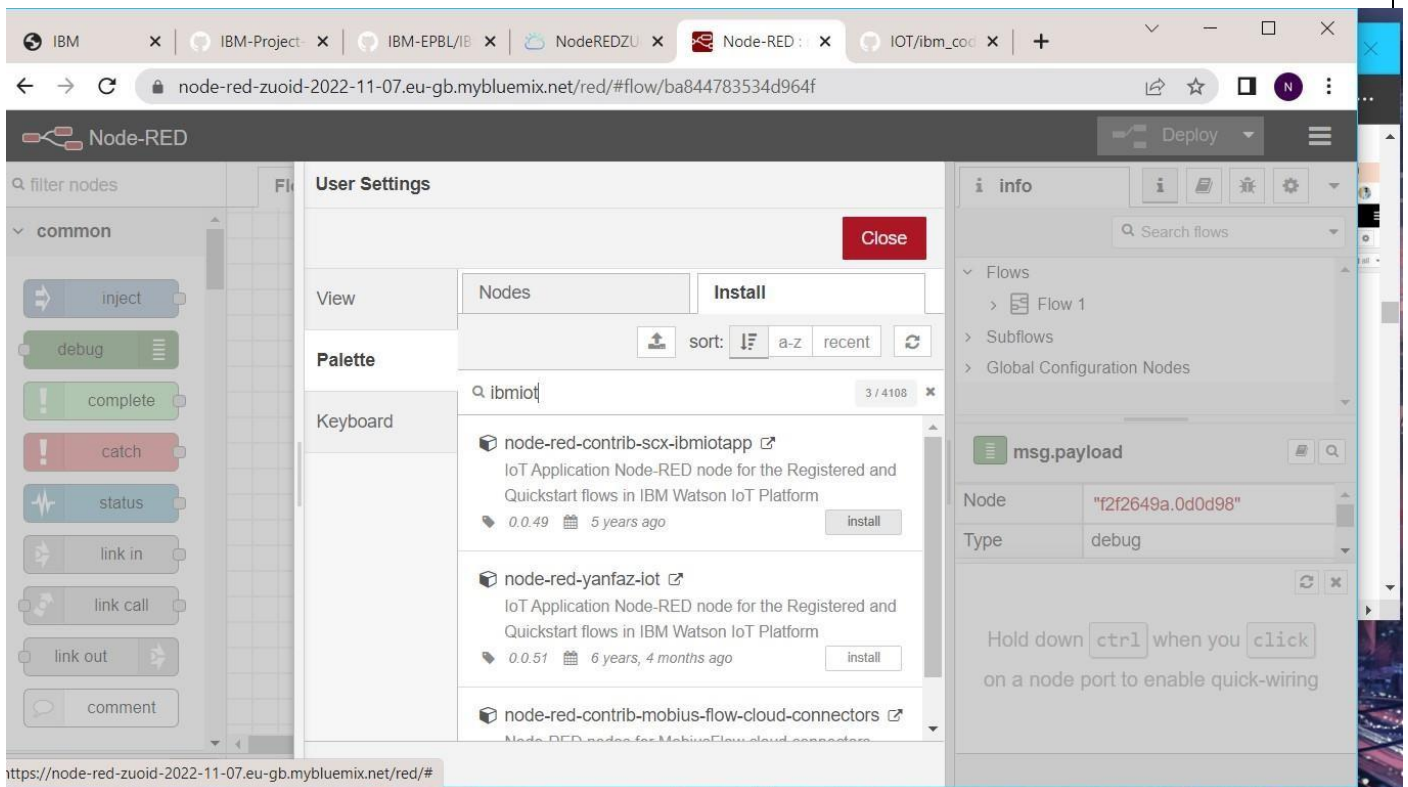


To install IBM nodes in Node-red flow editor click on manage palette in the menu option which is on the top-right of the screen.

○



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



Search for IBM nodes in the filter nodes section

The screenshot shows the Node-RED web interface in a browser. The browser's address bar displays the URL `node-red-zuoid-2022-11-07.eu-gb.mybluemix.net/red/#`. The interface includes a top navigation bar with a 'Deploy' button. On the left, a search bar contains the text 'ibm'. Below the search bar, the 'input' section lists the 'ibmiot in' node, and the 'output' section lists the 'ibmiot out' node. The central workspace is a grid for building flows. On the right, the 'info' sidebar is open, showing a search bar and a list of flows. Below the list, the 'IBM IoT' section is expanded, displaying the 'Node' as `"ae6ddad1b995b021"` and the 'Type' as 'ibmiot in'. At the bottom of the sidebar, there is a tip: 'click and drag on a node port to move all of the attached wires or just the selected one'.

- 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

○

The screenshot shows the Node-RED web interface in a browser. The address bar displays the URL: `node-red-zuoid-2022-11-07.eu-gb.mybluemix.net/red/#`. The interface includes a left sidebar with a 'filter nodes' search bar and a list of common nodes: inject, debug, complete, catch, status, link in, link call, link out, and comment. The main workspace, labeled 'Flow 1', contains a single 'IBM IoT' node. The right sidebar shows the 'info' panel with a search bar and a list of flows (Flow 1, Flow 2, Flow 3) and subflows. Below this, the 'IBM IoT' node is selected, showing its properties: Node ID is '18a789e653486de9' and Type is 'ibmiot in'. A message at the bottom of the info panel states: 'Pressing enter will edit the first node in the current selection'.

Select API Key from Authentication in properties.

○ In API Key paste API Key, API Token and server name and update it

The screenshot shows the Node-RED web interface with the 'Edit ibmiot in node' dialog box open. The dialog has a 'Delete' button, a 'Cancel' button, and a 'Done' button. It contains a 'Properties' section with the following fields: 'Authentication' (set to 'API Key'), 'API Key' (set to '857a9d97b49f4307'), 'Input Type' (set to 'Device Event'), 'Device Type' (set to 'All or TestDeviceType'), 'Device Id' (set to 'All or 2022'), 'Event' (set to 'All or +'), and 'Format' (set to 'All or json'). The 'Enabled' checkbox is checked. The background shows the same Node-RED interface as the previous screenshot, with the 'info' panel on the right displaying the 'IBM IoT' node properties.

○

- 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, a search bar, and a user profile section with the email 310819106054@smartinternz.com and ID: q26y5w. The main content area is titled 'IBM Cloud Apps' and features a 'Generate API Key' button. Below this, a table lists API keys. The first key is 'a-q26y5w-q7sakhzog4' with the description 'API Key for the device...' and role 'Standard Application'. A modal window titled 'API Key Information' is open, displaying details for the selected key:

API Key Information		Access Control/Permissions	
Key	a-q26y5w-q7sakhzog4	Last Edited By	-
Description	API Key for the device simulator	Expires	Never
Date Added	Nov 9, 2022 9:34 PM		
Last Update	Nov 9, 2022 9:34 PM		

At the bottom of the modal, it states '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.

o

The screenshot shows the Node-RED web interface in a browser. The top bar includes a 'Deploy' button. On the left, the 'common' node palette is visible. The central workspace, labeled 'Flow 1', contains a flow with two nodes: 'IBM IoT' (blue) and 'msg.payload' (green), connected by a line. The 'IBM IoT' node has a 'connected' status indicator. On the right, the 'debug' tab is active, showing a log of messages. The messages are JSON objects containing temperature and humidity data, along with timestamps and node IDs.

```
graph LR;
    subgraph Flow 1
        direction LR
        IoT[IBM IoT] --- Payload[msg.payload]
    end
```

debug

- 11/9/2022, 12:44:38 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : Object
▶ { temperature: 75, humidity: 97 }
- 11/9/2022, 12:44:40 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : Object
▶ { temperature: 44, humidity: 89 }
- 11/9/2022, 12:44:43 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : Object
▶ { temperature: 93, humidity: 96 }
- 11/9/2022, 12:44:45 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : Object
▶ { temperature: 79, humidity: 15 }
- ▶ { temperature: 59, humidity: 99 }

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

O

IBM x Round x IBM-Pr x IBM-EP x Service x IBM W x Node-I x IOT/ibr x +

node-red-zuoid-2022-11-07.eu-gb.mybluemix.net/red/#flow/ba844783534d964f

Node-RED Deploy

filter nodes

Flow 1 Flow 2

common

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

Flow 1

IBM IoT

connected

msg.payload

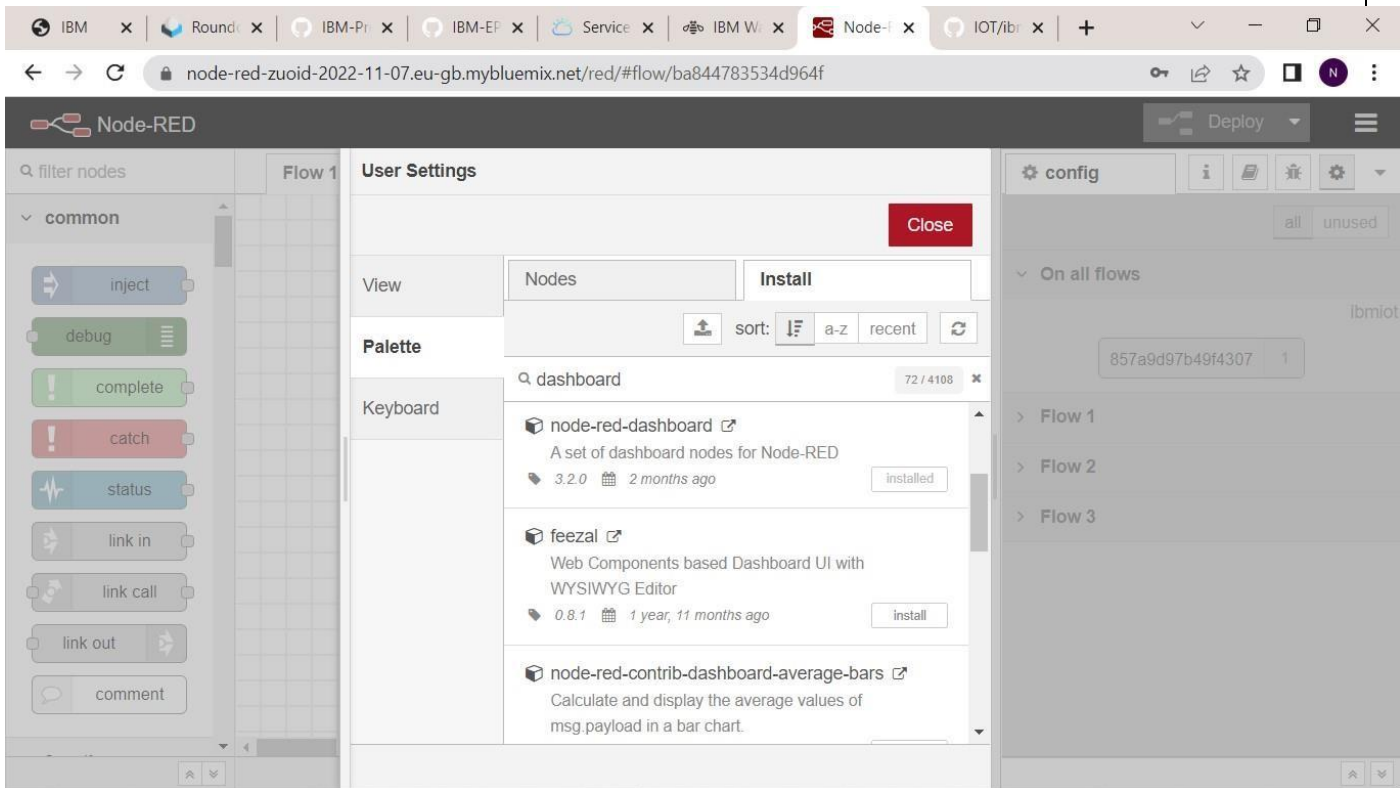
debug

all nodes

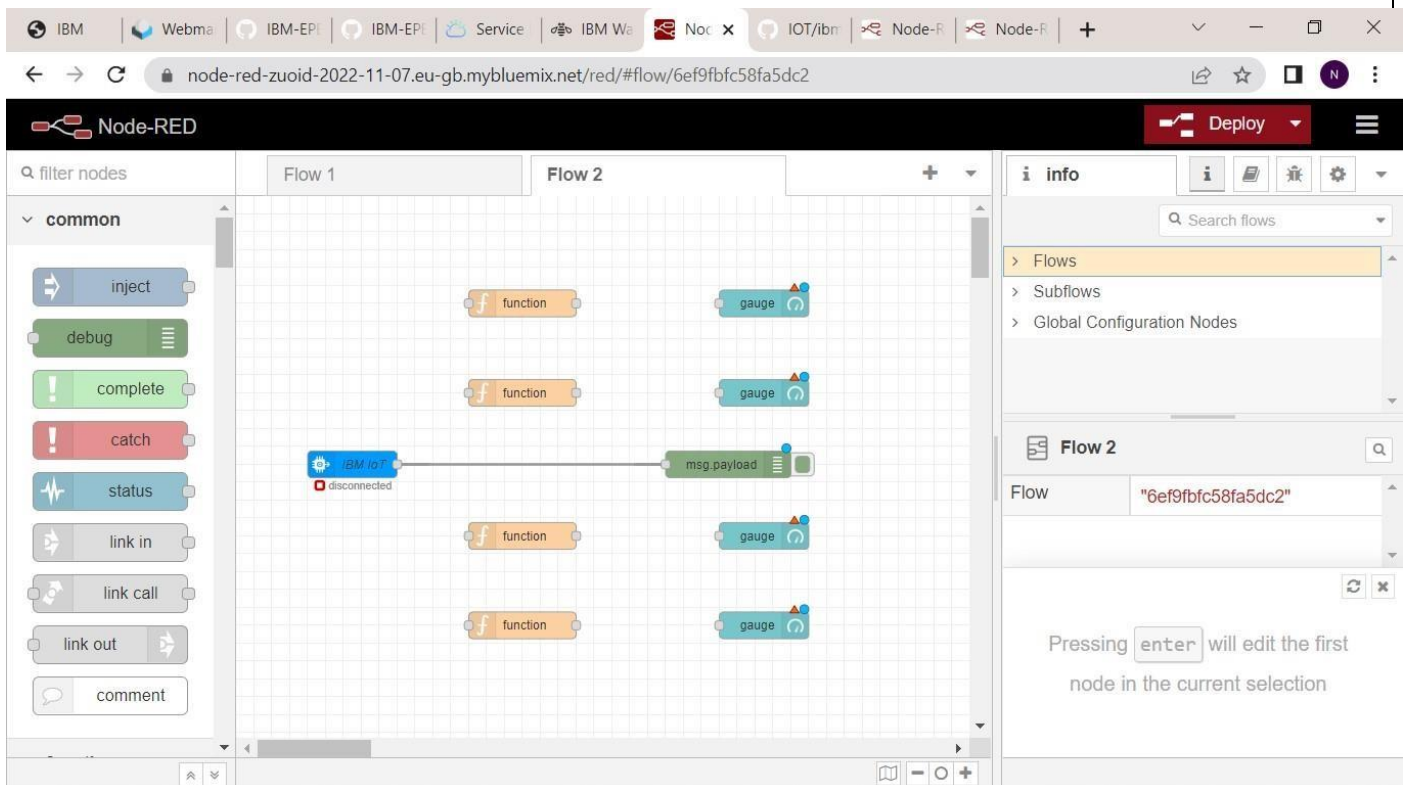
all

```
{ temperature: 66, humidity: 75 }
11/9/2022, 12:46:00 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : Object
{ temperature: 14, humidity: 73 }
11/9/2022, 12:46:02 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : Object
{ temperature: 101, humidity: 5 }
11/9/2022, 12:46:05 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : Object
{ temperature: 100, humidity: 92 }
11/9/2022, 12:46:06 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : Object
{ temperature: 17, humidity: 8 }
```

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

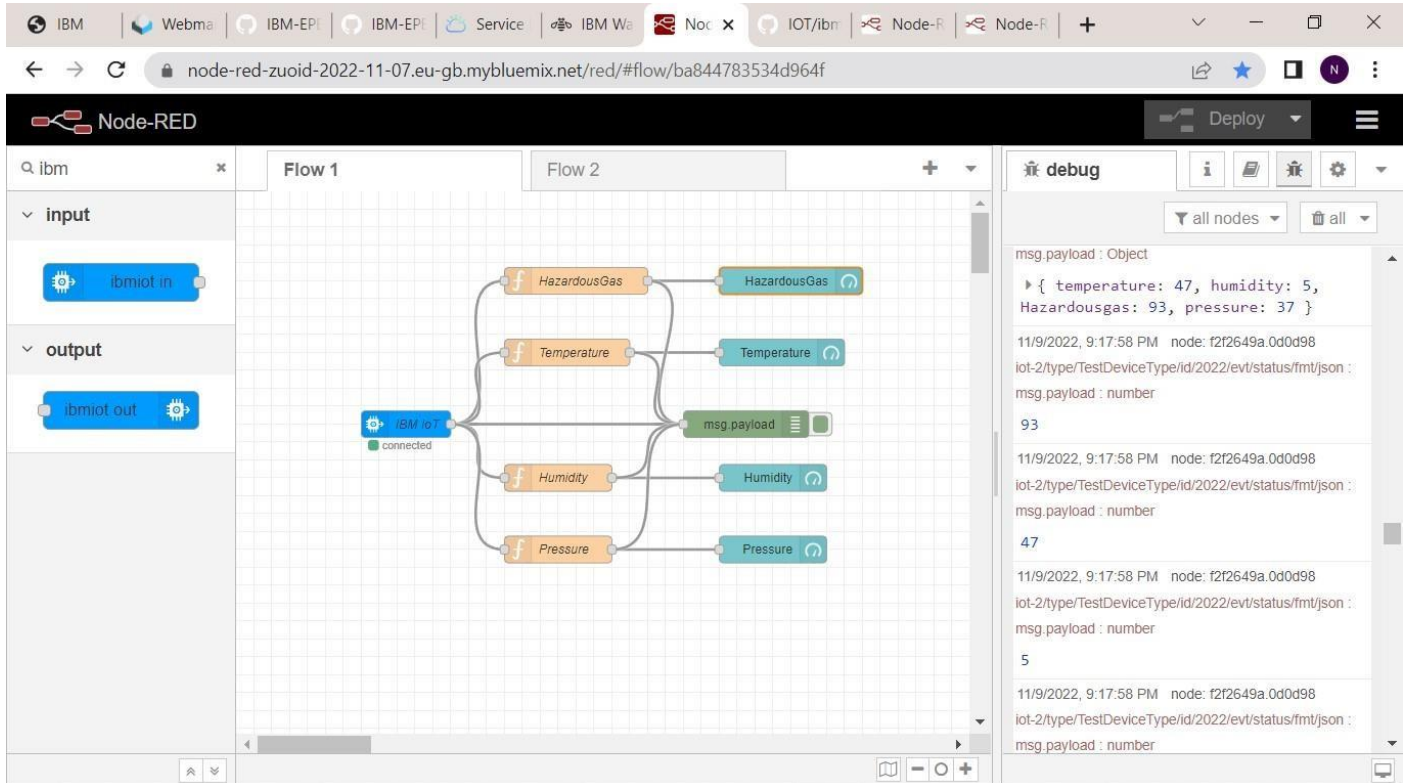


- Drag and place the function node and gauge node in the flow editor to separate the temperature and humidity value



○

- 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

IBM Webma IBM-EPI IBM-EPI Service IBM Wa Noc x IOT/ibm Node-R Node-R +

node-red-zuoid-2022-11-07.eu-gb.mybluemix.net/red/#flow/ba844783534d964f

Node-RED

Flow 1 Flow 2

input

ibmiot in

output

ibmiot out

Flow 1

ibmiot in

HazardousGas

Temperature

Humidity

Pressure

HazardousGas

Temperature

Humidity

Pressure

msg.payload

debug

all nodes

11/9/2022, 9:18:41 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : number
40

11/9/2022, 9:18:42 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : number
41

11/9/2022, 9:18:43 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : Object
{ temperature: 93, humidity: 81, Hazardousgas: 96, pressure: 97 }

11/9/2022, 9:18:44 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : number
96

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

IBM Webma IBM-EPI IBM-EPI Service IBM Wa Noc x IOT/ibm Node-R Node-R +

node-red-zuoid-2022-11-07.eu-gb.mybluemix.net/red/#flow/ba844783534d964f

Node-RED

Flow 1

input

ibmiot in

output

ibmiot out

Edit gauge node

Delete Cancel Done

Properties

Group [Hazardous gas] Gas leakage

Size auto

Type Compass

Label Pressure

Value format {{value}}

Units %

Range min 0 max 100

Enabled

debug

all nodes

11/9/2022, 9:19:39 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : number
18

11/9/2022, 9:19:39 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : number
110

11/9/2022, 9:19:39 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : number
80

11/9/2022, 9:19:39 PM node: f2f2649a.0d0d98
iot-2/type/TestDeviceType/id/2022/evt/status/fmt/json :
msg.payload : number
88

11/9/2022, 9:19:41 PM node: f2f2649a.0d0d98

○

○ After editing the nodes, deploy it

The screenshot displays the Node-RED web application interface. The top navigation bar includes links for IBM, Webmail, IBM-EPI, Service, and IBM Watson. The browser address bar shows the URL: `node-red-zuoid-2022-11-07.eu-gb.mybluemix.net/red/#flow/ba844783534d964f`. The main workspace is titled "Flow 1" and contains a flow diagram. The flow starts with an "IBM IoT" node (labeled "connected") which connects to four function nodes: "HazardousGas", "Temperature", "Humidity", and "Pressure". Each function node is connected to a corresponding output node: "HazardousGas", "Temperature", "Humidity", and "Pressure". The left sidebar shows the "input" and "output" node categories. The right sidebar shows the "debug" console with a list of messages. The messages are JSON objects containing the following data:

```
{
  "temperature": 93,
  "humidity": 81,
  "Hazardousgas": 96,
  "pressure": 97
}
```

The debug console also shows the "msg.payload" for each message, which is a number (40, 41, 96).

RESULT:

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

