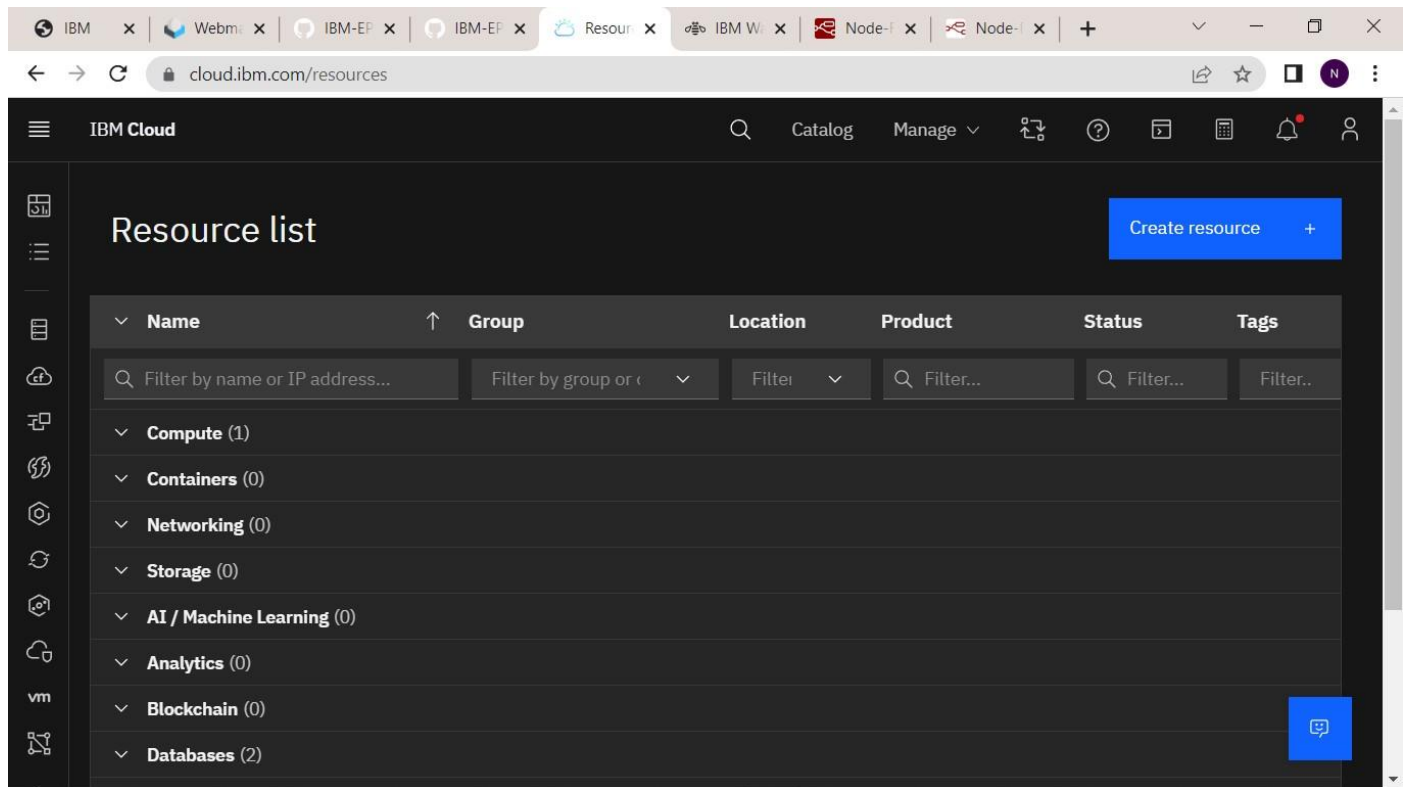


Creating a Node-Red Web Application to view data in Separate Numerical form

TEAM ID	PNT2022TMID00945
PROJECT NAME	GAS LEAKAGE MONITORING AND ALERTING SYSTEM FOR INDUSTRIES

- In IBM cloud dashboard, click on Cloud Foundry apps



The screenshot shows the IBM Cloud dashboard interface. The browser address bar displays 'cloud.ibm.com/resources'. The main heading is 'Resource list', with a 'Create resource +' button in the top right. Below the heading is a table with columns: Name, Group, Location, Product, Status, and Tags. Each column has a search filter input. The table lists several resource categories with their counts: 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 table area.

Name	Group	Location	Product	Status	Tags
Filter by name or IP address...	Filter by group or type...	Filter...	Filter...	Filter...	Filter...
Compute (1)					
Containers (0)					
Networking (0)					
Storage (0)					
AI / Machine Learning (0)					
Analytics (0)					
Blockchain (0)					
Databases (2)					

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

The screenshot displays the IBM Cloud 'Resource list' interface. At the top, there's a navigation bar with the IBM Cloud logo and a search bar. Below this, a table lists resources. The table has columns: Name, Group, Location, Product, Status, and Tags. A resource named 'Node RED ZUOID 2022-11-07' is highlighted, and a tooltip shows its full name. The resource is in the 'Nandha kumar / dev' group, located in 'London', using 'Node.js' product, and has a status of 'Started'.

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.

The screenshot shows the IBM Cloud console interface for a resource named 'Node RED ZUOID 2022-11-07'. The status is 'Running'. The left sidebar contains navigation links: 'Overview' (selected), 'Runtime', 'Connections', 'Logs', 'API Management', and 'Autoscaling'. The main content area displays the resource name, status, and a 'Visit App URL' button. Below this, there's a section for 'Instances' showing 1 instance running with 100% health, and a 'Runtime' section showing a donut chart for memory allocation (256 MB total, 1.75 GB available).



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

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.

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

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

Node-RED

filter nodes

common

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

Flow 1 Flow 2

info

Search flows

Flows

- Flow 1
- Flow 2

Subflows

Global Configuration Nodes

Flow 2

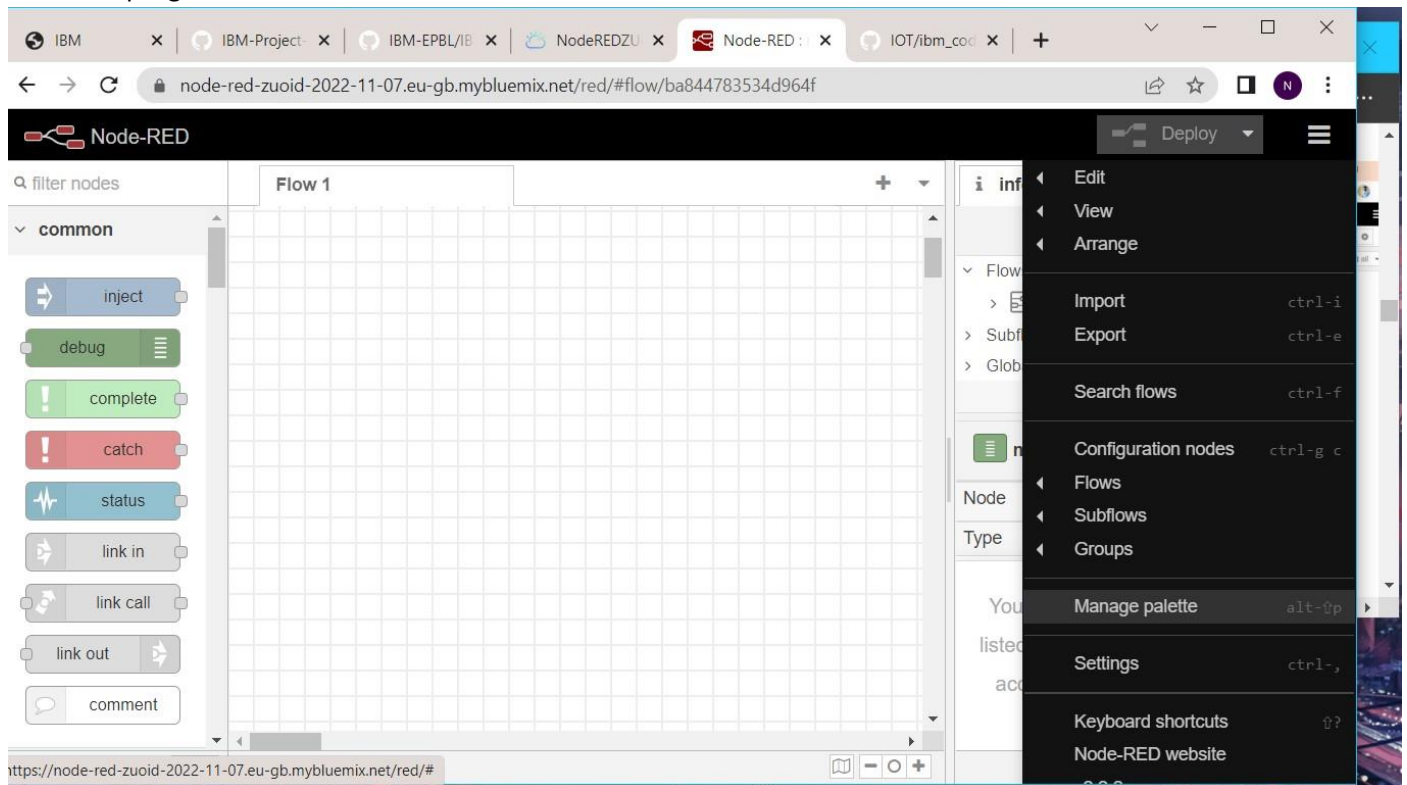
Flow "6ef9bfc58fa5dc2"

ctrl click in the workspace to open the quick-add dialog

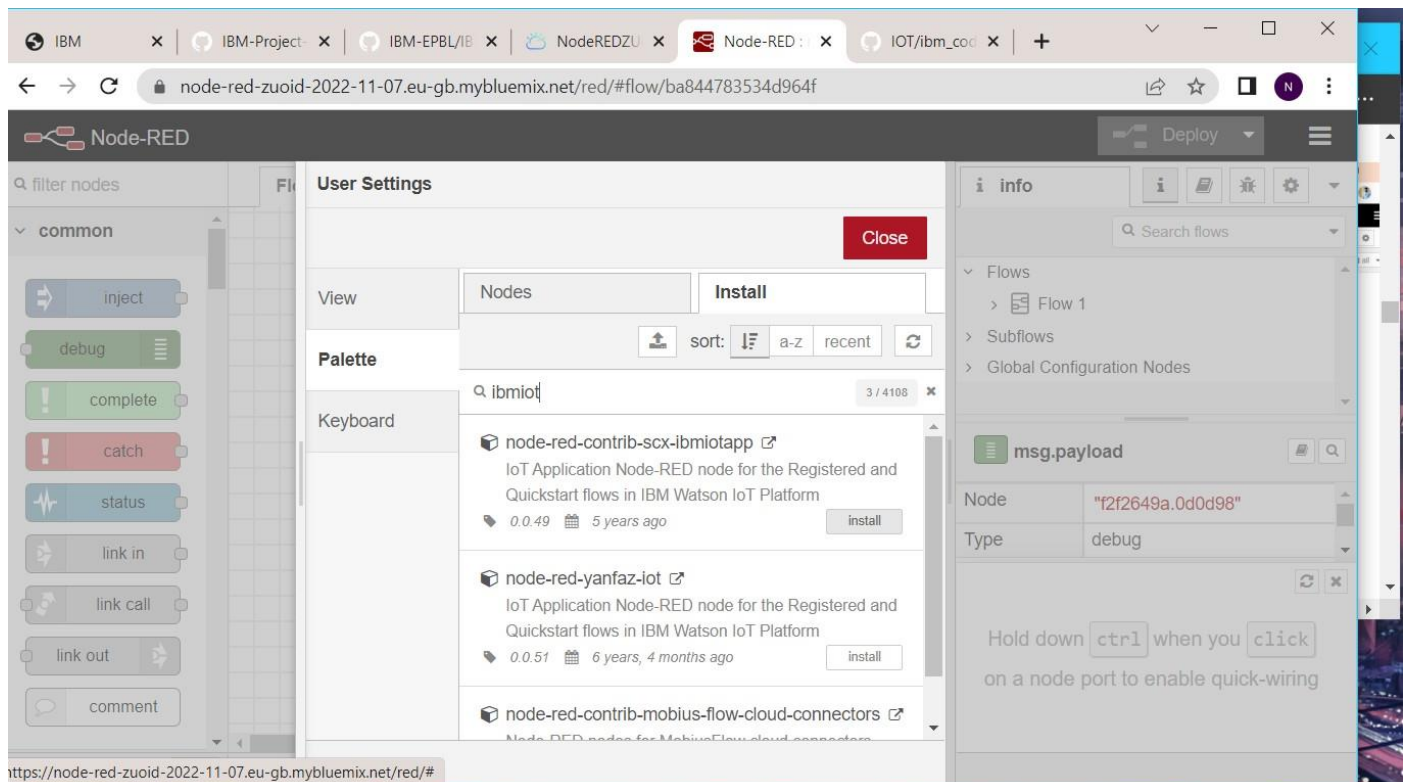


○

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 top bar includes a 'Deploy' button and a menu icon. The left sidebar has a search bar with 'ibm' entered. Below the search bar, the 'input' and 'output' categories are expanded, showing 'ibmiot in' and 'ibmiot out' nodes respectively. The main workspace is a grid with 'Flow 1' and 'Flow 2' tabs. The right sidebar shows the 'info' panel with a 'Search flows' dropdown and a list of flows. Below this, the 'IBM IoT' section is active, displaying the node type 'ibmiot in' and its ID 'ae6ddad1b995b021'. A tooltip at the bottom right of the info panel reads: '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. The left sidebar has a search bar with 'filter nodes' entered. Below the search bar, the 'common' category is expanded, showing various nodes including 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', and 'comment'. The main workspace is a grid with 'Flow 1' and 'Flow 2' tabs. The right sidebar shows the 'info' panel with a 'Search flows' dropdown and a list of flows. Below this, the 'IBM IoT' section is active, displaying the node type 'ibmiot in' and its ID '18a789e653486de9'. A tooltip at the bottom right of the info panel reads: '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 in a browser. The browser's address bar displays the URL: `node-red-zuoid-2022-11-07.eu-gb.mybluemix.net/red/#`. The interface is divided into three main sections: 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, comment); a central workspace titled 'Edit ibmiot in node' with 'Delete', 'Cancel', and 'Done' buttons; and a right sidebar with an 'info' tab and a list of flows (Flow 1, Flow 2, Flow 3, Subflows, Global Configuration Nodes). The 'Properties' section in the central workspace is expanded, showing the following configuration:

- Authentication:** API Key
- API Key:** 857a9d97b49f4307
- Input Type:** Device Event
- Device Type:** All or TestDeviceType
- Device Id:** All or 2022
- Event:** All or +
- Format:** All or json

At the bottom of the properties section, there is an 'Enabled' checkbox. The right sidebar also shows the 'IBM IoT' section with the following details:

- Node:** "18a789e653486de9"
- Type:** ibmiot in

Below the details, there is a note: 'Switch flow tabs with ctrl-[and ctrl-]'.

- 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 text 'IBM Watson IoT Platform'. The user's email '310819106054@smartinternz.com' and ID 'ID: q26y5w' are displayed in the top right corner. The main content area is titled 'Browse 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'. The 'API Key Information' section is expanded, showing details for the selected key.

Key	Description	Role	Expires
a-q26y5w-q7sakhzog4	API Key for the device...	Standard Application	-

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		

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 web interface in a browser. The flow editor has two tabs: 'Flow 1' and 'Flow 2'. In 'Flow 1', there is a flow with an 'IBM IoT' node (blue icon with a gear) connected to a 'msg.payload' node (green icon with a list). The 'IBM IoT' node is labeled 'connected'. The left sidebar shows a 'common' category with various nodes like 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', and 'comment'. The right sidebar shows the 'debug' tab with a list of messages. The messages are JSON objects containing 'temperature' and 'humidity' values. The messages are as follows:

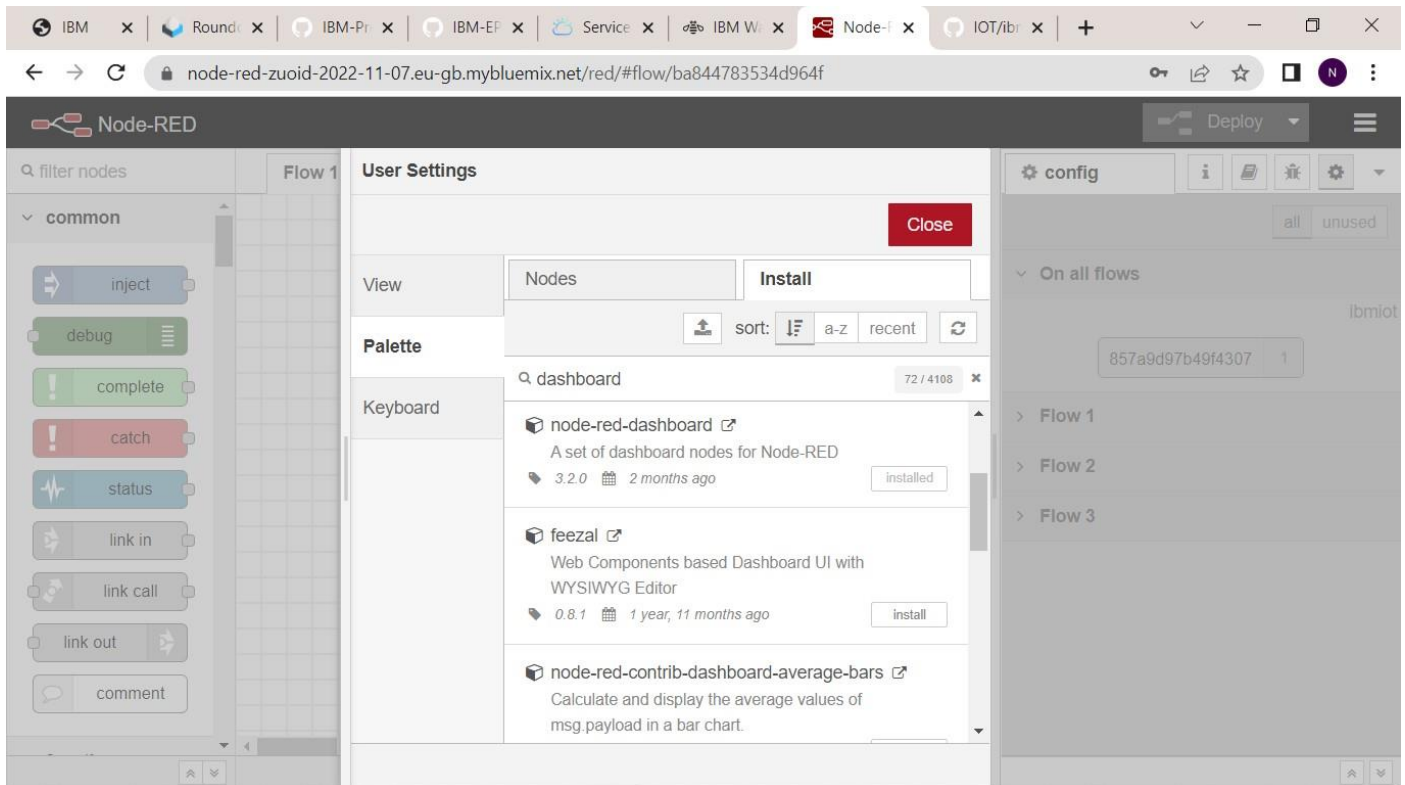
Time	Node ID	msg.payload
11/9/2022, 12:44:38 PM	node: f2f2649a.0d0d98	{ temperature: 75, humidity: 97 }
11/9/2022, 12:44:40 PM	node: f2f2649a.0d0d98	{ temperature: 44, humidity: 89 }
11/9/2022, 12:44:43 PM	node: f2f2649a.0d0d98	{ temperature: 93, humidity: 96 }
11/9/2022, 12:44:45 PM	node: f2f2649a.0d0d98	{ temperature: 79, humidity: 15 }
11/9/2022, 12:44:45 PM	node: f2f2649a.0d0d98	{ 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

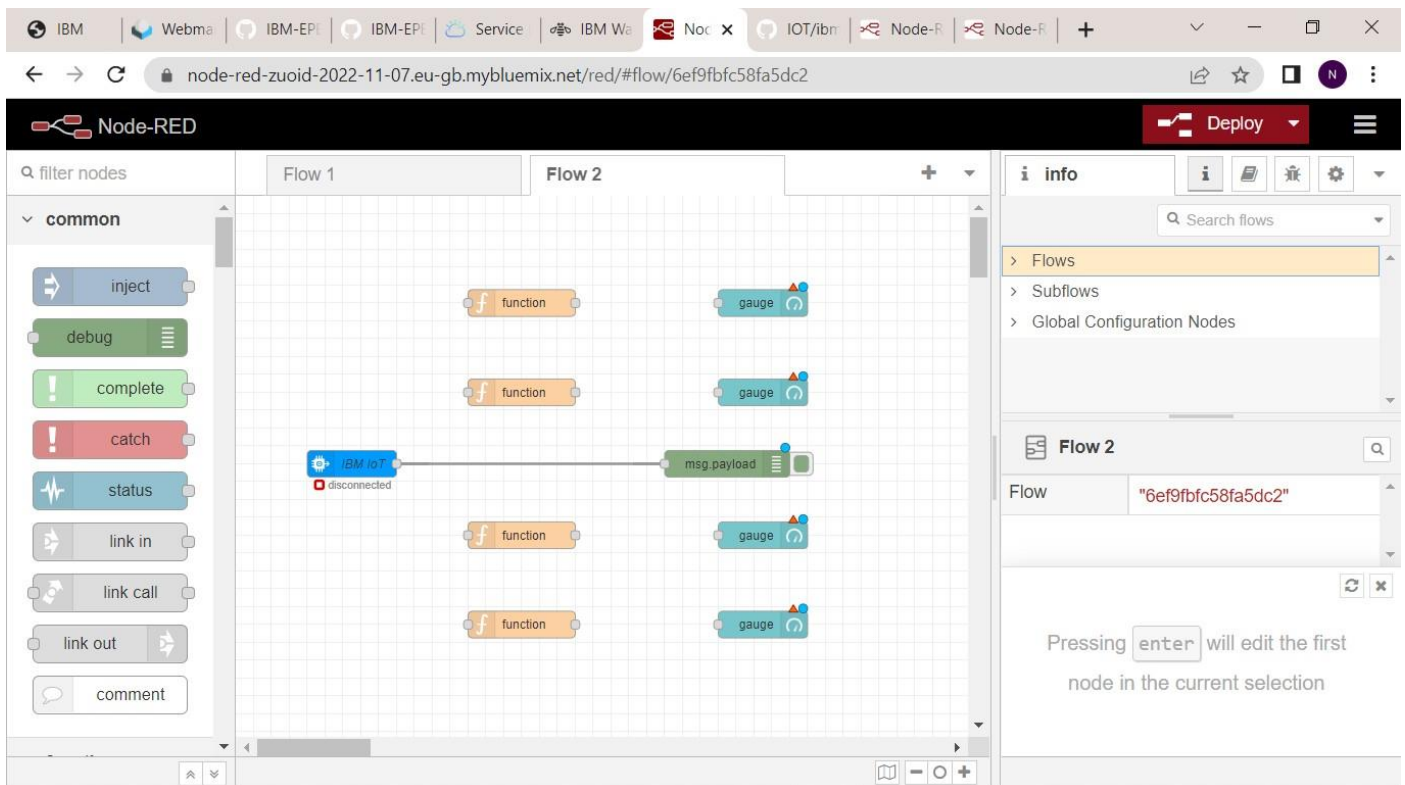
The screenshot shows the same Node-RED interface as the previous one, but with a red box highlighting the 'debug' tab on the right. The 'debug' tab now shows a new message with temperature: 66 and humidity: 75. The messages are as follows:

Time	Node ID	msg.payload
11/9/2022, 12:46:00 PM	node: f2f2649a.0d0d98	{ temperature: 66, humidity: 75 }
11/9/2022, 12:46:02 PM	node: f2f2649a.0d0d98	{ temperature: 14, humidity: 73 }
11/9/2022, 12:46:05 PM	node: f2f2649a.0d0d98	{ temperature: 101, humidity: 5 }
11/9/2022, 12:46:06 PM	node: f2f2649a.0d0d98	{ temperature: 100, humidity: 92 }
11/9/2022, 12:46:06 PM	node: f2f2649a.0d0d98	{ 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

The screenshot shows the Node-RED web interface in a browser. The left sidebar contains an 'input' section with 'ibmiot in' and an 'output' section with 'ibmiot out'. The main workspace shows 'Flow 1' with an 'IBM IoT' node connected to four function nodes: 'HazardousGas', 'Temperature', 'Humidity', and 'Pressure'. Each function node is connected to a corresponding output node on the right. The right sidebar shows the 'debug' console with a list of messages. The current message is an object: `{ temperature: 47, humidity: 5, Hazardousgas: 93, pressure: 37 }`.

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

This screenshot shows the same Node-RED interface, but with the 'msg.payload' node selected in the workspace. The debug console shows a new message object: `{ temperature: 93, humidity: 81, Hazardousgas: 96, pressure: 97 }`. The values for temperature, humidity, hazardous gas, and pressure have all increased significantly from the previous state.

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

IBM | Webma | IBM-EP | IBM-EP | 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

Deploy

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

IBM | Webma | IBM-EP | IBM-EP | 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

Deploy

Flow 1

Flow 2

input

ibmiot in

output

ibmiot out

Flow 2

ibmiot in

HazardousGas

Temperature

Humidity

Pressure

msg.payload

HazardousGas

Temperature

Humidity

Pressure

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

11/9/2022, 9:18:45 PM node: f2f2649a.0d0d98

RESULT:

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

.