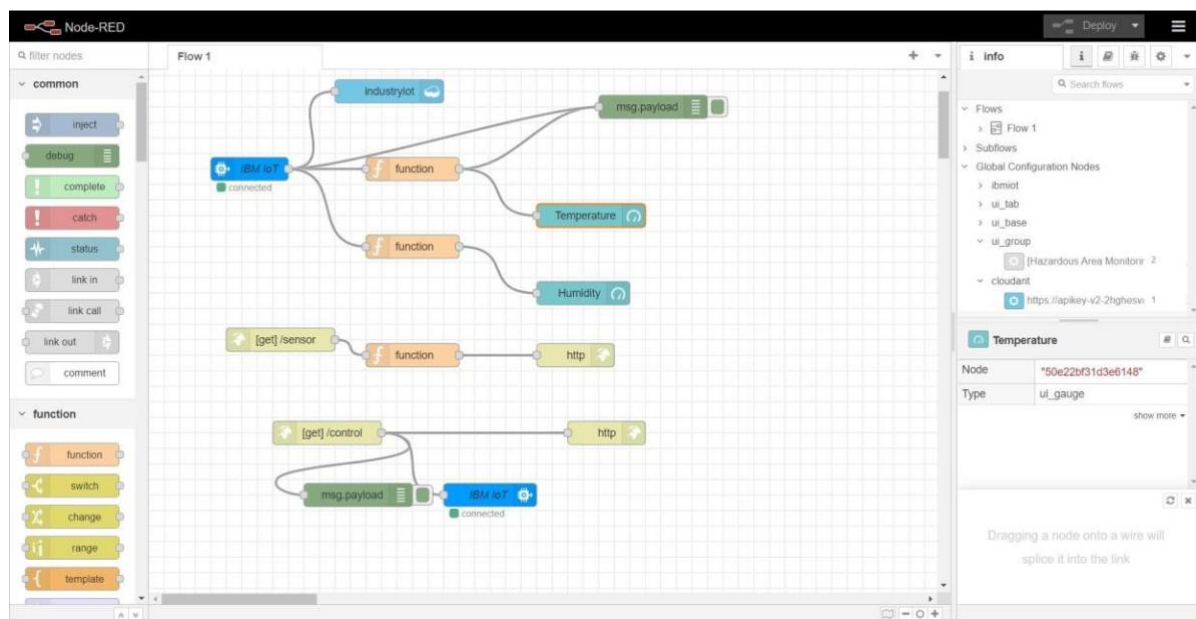


# Configure The Application To Receive The Data From Cloud

Team Id - PNT2022TMID20013  
Maximum Marks - 4 Marks

Node red flow created to get values



## Configuring function to fetch the desired value

The screenshot shows the Node-RED web interface. On the left, a flow named 'Flow 1' is visible. It starts with an 'IBM IoT' node (connected), which branches into three 'function' nodes. The top 'function' node is selected, and its configuration is shown in the 'Edit function node' dialog on the right. The dialog has tabs for 'Properties', 'Setup', 'On Start', 'On Message', and 'On Stop'. The 'On Message' tab is active, showing the following JavaScript code:

```
1 msg.payload = msg.payload.temp;  
2 global.set('t',msg.payload)  
3 return msg;
```

At the bottom of the dialog, there is an 'Enabled' checkbox which is currently checked.

This screenshot is similar to the one above, showing the same flow and the 'Edit function node' dialog. However, the 'On Message' tab now contains the following JavaScript code:

```
1 msg.payload = msg.payload.humid;  
2 global.set('h',msg.payload)  
3 return msg;
```

The 'Enabled' checkbox at the bottom remains checked.

## App Blocks to render the values and display it in app

The screenshot displays the MIT App Inventor web interface for a project named "IoT\_Interface". The interface is divided into several sections:

- Top Bar:** Includes navigation links like "Projects", "Connect", "Build", "Settings", and "Help". It also shows the user's profile "jagangoguria@gmail.com".
- Left Panel (Blocks):** A sidebar containing various built-in blocks categorized under "Built-in", "Screen1", and "Media".
- Right Panel (Viewer):** The main workspace where the app's logic is built using a block-based programming language. The logic includes:
  - A "when Clock1.Timer" block triggering a "do" block that sets "Web1.Uri" to "http://159.122.177.234:30241/sensor" and calls "Web1.Get".
  - A "when Web1.GetTest" block with a "do" block that processes the response. It uses "lock up in pairs" to extract "temperature" and "humidity" from the response content, then calls "Web1.JsonTextDecode" and "Web1.JsonText" to decode and display the data in "textBox1" and "textBox2".
  - Two "when Button1.Click" and "when Button2.Click" blocks that trigger "do" blocks to set "Web2.Uri" to "http://159.122.177.234:30241/control?command=on" and "http://159.122.177.234:30241/control?command=off" respectively, and call "Web2.Get".
- Bottom Bar:** Contains a "Show Warnings" button and a "Project Policy and Terms of Use" link.