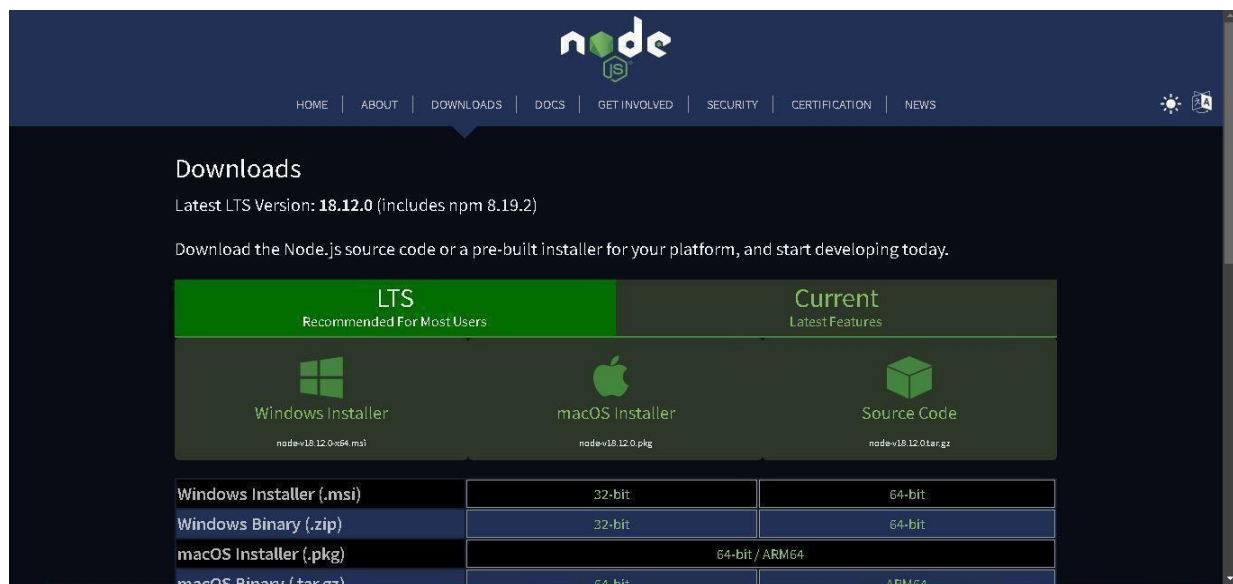


# SPRINT 2

<b>Date</b>	November 10, 2022
<b>Team ID</b>	PNT2022TMID26105
<b>Project Name</b>	Real-Time River Water Quality Monitoring and Control System
<b>Maximum Mark</b>	

**The following steps are involved:**

**STEP 1:** Download and Install node.js.



**STEP 2:** Setup node.js and configure command prompt for error check. Open node-red from the generated link.

```
npm
C:\Users\Ajay>npm install -g --unsafe-perm node-red
npm WARN deprecated @types/keyv@4.2.0: This is a stub types definition. keyv provides its own type definitions, so you do not need this installed.

added 292 packages, and audited 293 packages in 5m

39 packages are looking for funding
  run `npm fund` for details

5 vulnerabilities (4 low, 1 moderate)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

C:\Users\Ajay>node-red
B Nov 14:35:28 - [info]

Welcome to Node-RED
=====

B Nov 14:35:29 - [info] Node-RED version: v3.0.2
B Nov 14:35:29 - [info] Node.js version: v18.12.0
B Nov 14:35:29 - [info] Windows_NT 10.0.19043 x64 LE
B Nov 14:35:44 - [info] Loading palette nodes
```

### STEP 3: Generating API key and Authentication token.

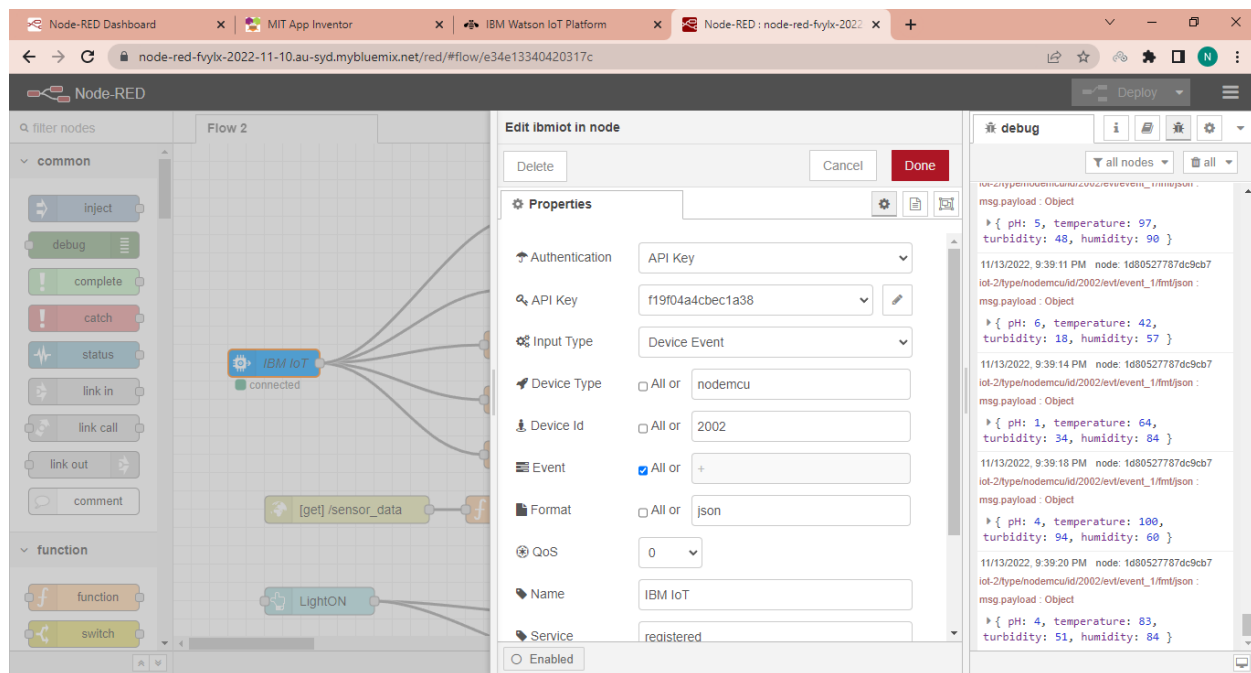
The screenshot shows the IBM Watson IoT Platform dashboard. The main heading is "The API key has been added." Below this, a warning message states: "Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the API key to generate a new authentication token." The dashboard is divided into two columns: "Generated Details" and "API Key Information".

Generated Details		API Key Information	
API Key	a-2208jk-3wkprnmnkx	Description	iot_sensor
Authentication Token	nWAYji(HhF0cDAxN0	Role	Standard Application
		Expires	Never

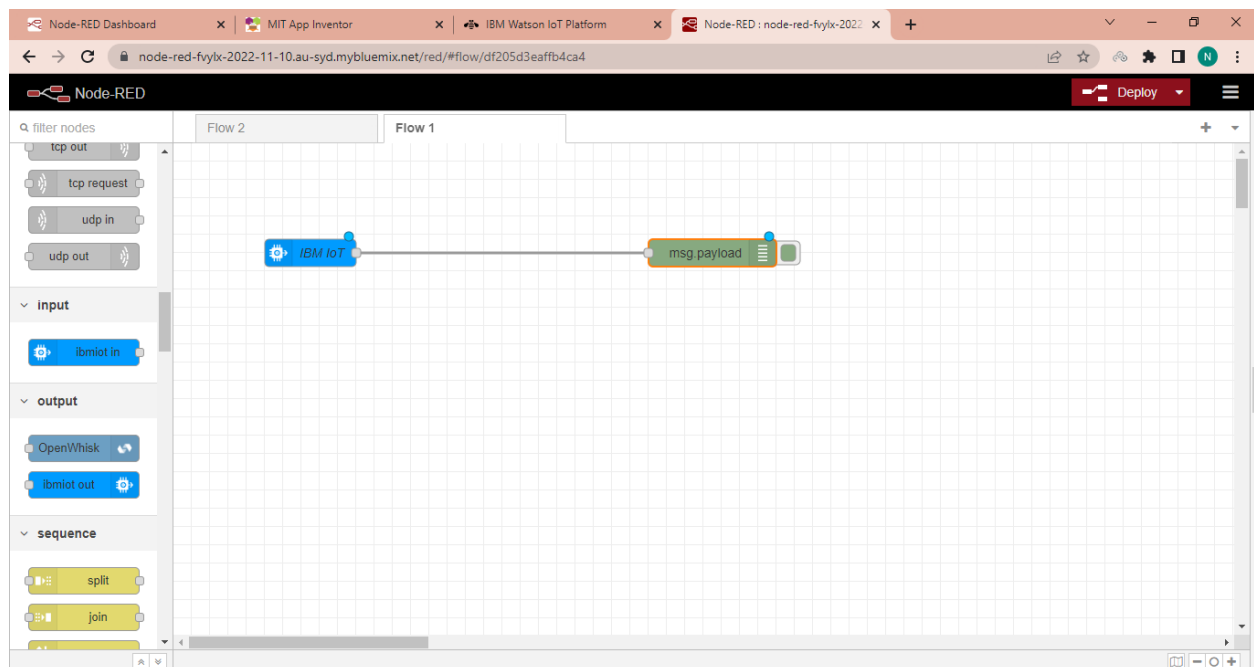
Below the table, a warning icon and text state: "Make a note of the generated authentication token. Lost authentication tokens cannot be recovered. If you lose the token, you must reregister the API to generate a new token." At the bottom right of the main content area are three buttons: "View API Key", "Add Another", and "Close".

Below the main content area, there is a section titled "Browse API Keys" with a status indicator "1 Simulation running".

### STEP 4: Edit Ibmiot in node.



**STEP 5:** Connect Ibmiot in and debug 1 and deploy.



## STEP 6: Edit gauge node (here the gauge nodes are named as Temperature, pH and Turbidity).

The screenshot shows the Node-RED web interface. In the center, the 'Edit gauge node' dialog is open for a node named 'pH'. The properties are configured as follows:

- Group: [Home] iot\_sensor\_Analytics
- Size: auto
- Type: Gauge
- Label: pH
- Value format: {{value}}
- Units: units
- Range: min 0, max 7
- Colour gradient: A gradient bar with green, yellow, and red segments.
- Sectors: 0, optional, optional, 7
- Class: Optional CSS class name(s) for widget
- Enabled: ☒

The debug console on the right shows a series of JSON messages received from the IoT platform, including pH, temperature, and turbidity readings.

The screenshot shows the Node-RED web interface. In the center, the 'Edit gauge node' dialog is open for a node named 'temperature'. The properties are configured as follows:

- Group: [Home] iot\_sensor\_Analytics
- Size: auto
- Type: Gauge
- Label: temperature
- Value format: {{value}}
- Units: Degree celcius
- Range: min 0, max 100
- Colour gradient: A gradient bar with green, yellow, and red segments.
- Sectors: 0, optional, optional, 100
- Class: Optional CSS class name(s) for widget
- Enabled: ☒

The debug console on the right shows a series of JSON messages received from the IoT platform, including pH, temperature, and turbidity readings.

Node-RED Dashboard | MIT App Inventor | IBM Watson IoT Platform | Node-RED : node-red-fylix-2022

node-red-fylix-2022-11-10-au-syd.mybluemix.net/red/#flow/e34e13340420317c

### Node-RED

filter nodes | Flow 2

tcp out | tcp request | udp in | udp out

input: ibmiot in

output: OpenWhisk | ibmiot out

sequence: split | join

IBM IoT (connected) | [get] /sensor\_data | LightON

#### Edit gauge node

Delete | Cancel | Done

##### Properties

Group: [Home] iot\_sensor\_Analytics

Size: auto

Type: Gauge

Label: Humidity

Value format: {{value}}

Units: %

Range: min 0 max 100

Colour gradient: [Green] [Yellow] [Red]

Sectors: 0 ... optional ... optional ... 100

Class: Optional CSS class name(s) for widget

☐ Enabled

#### debug

all nodes | all

```
Object
  > { pH: 0, temperature: 84, turbidity: 19, humidity: 77 }
11/13/2022, 9:43:00 PM node: 1d80527787d9cb7
iot-2type/nodemcuId/2002/ev/1event_1/fmt/json : msg.payload :
Object
  > { pH: 5, temperature: 77, turbidity: 67, humidity: 47 }
11/13/2022, 9:43:03 PM node: 1d80527787d9cb7
iot-2type/nodemcuId/2002/ev/1event_1/fmt/json : msg.payload :
Object
  > { pH: 5, temperature: 34, turbidity: 62, humidity: 41 }
11/13/2022, 9:43:06 PM node: 1d80527787d9cb7
iot-2type/nodemcuId/2002/ev/1event_1/fmt/json : msg.payload :
Object
  > { pH: 2, temperature: 6, turbidity: 80, humidity: 19 }
11/13/2022, 9:43:09 PM node: 1d80527787d9cb7
iot-2type/nodemcuId/2002/ev/1event_1/fmt/json : msg.payload :
Object
  > { pH: 6, temperature: 43, turbidity: 97, humidity: 80 }
```

Node-RED Dashboard | MIT App Inventor | IBM Watson IoT Platform | Node-RED : node-red-fylix-2022

node-red-fylix-2022-11-10-au-syd.mybluemix.net/red/#flow/e34e13340420317c

### Node-RED

filter nodes | Flow 2

tcp out | tcp request | udp in | udp out

input: ibmiot in

output: OpenWhisk | ibmiot out

sequence: split | join

IBM IoT (connected) | [get] /sensor\_data | LightON

#### Edit gauge node

Delete | Cancel | Done

##### Properties

Group: [Home] iot\_sensor\_Analytics

Size: auto

Type: Gauge

Label: Turbidity

Value format: {{value}}

Units: NTU

Range: min 0 max 100

Colour gradient: [Green] [Yellow] [Red]

Sectors: 0 ... optional ... optional ... 100

Class: Optional CSS class name(s) for widget

☐ Enabled

#### debug

all nodes | all

```
Object
  > { pH: 5, temperature: 54, turbidity: 68, humidity: 55 }
11/13/2022, 9:43:18 PM node: 1d80527787d9cb7
iot-2type/nodemcuId/2002/ev/1event_1/fmt/json : msg.payload :
Object
  > { pH: 5, temperature: 99, turbidity: 24, humidity: 53 }
11/13/2022, 9:43:21 PM node: 1d80527787d9cb7
iot-2type/nodemcuId/2002/ev/1event_1/fmt/json : msg.payload :
Object
  > { pH: 3, temperature: 86, turbidity: 18, humidity: 45 }
11/13/2022, 9:43:24 PM node: 1d80527787d9cb7
iot-2type/nodemcuId/2002/ev/1event_1/fmt/json : msg.payload :
Object
  > { pH: 4, temperature: 8, turbidity: 23, humidity: 49 }
11/13/2022, 9:43:27 PM node: 1d80527787d9cb7
iot-2type/nodemcuId/2002/ev/1event_1/fmt/json : msg.payload :
Object
  > { pH: 3, temperature: 54, turbidity: 0, humidity: 33 }
```

## STEP 7: Simulated program to get the random values.

The screenshot displays the IBM Watson IoT Platform interface. On the left, a sidebar contains navigation icons. The main panel shows a table of devices with columns for Device ID, Status, and Device Type. Two devices are listed: 123456 (Disconnected, iotsura) and 2002 (Disconnected, nodemcu). The 'Recent Events' tab is selected for device 2002. On the right, a modal window titled 'Device Type: nodemcu' is open, showing the 'Events' configuration. It includes a 'New event type' button, a 'Send' button, a 'Schedule' dropdown set to 'Every Minute', and a 'Payload' editor. The payload is a JSON object with random values for pH, temperature, turbidity, and humidity.

Device ID	Status	Device Type
123456	Disconnected	iotsura
2002	Disconnected	nodemcu

```
{
  "pH": random(0, 7),
  "temperature": random(0, 100),
  "turbidity": random(0, 100),
  "humidity": random(0, 100)
}
```

## STEP 8: Generate debug message from IBM Watson IoT Platform and connect the nodes.

The screenshot shows the Node-RED interface with a flow named 'Flow 2'. The flow starts with an 'IBM IoT' node (connected), which branches into four function nodes labeled 'pH', 'temperature', 'humidity', and 'turbidity'. Each function node is connected to a corresponding output node (pH, temperature, humidity, turbidity). Below these, there are two more function nodes labeled 'function' and 'http', connected to 'LightON' and 'LightOFF' nodes. The debug console on the right shows a series of messages, including the payload from the 'IBM IoT' node and the output from the 'function' node.

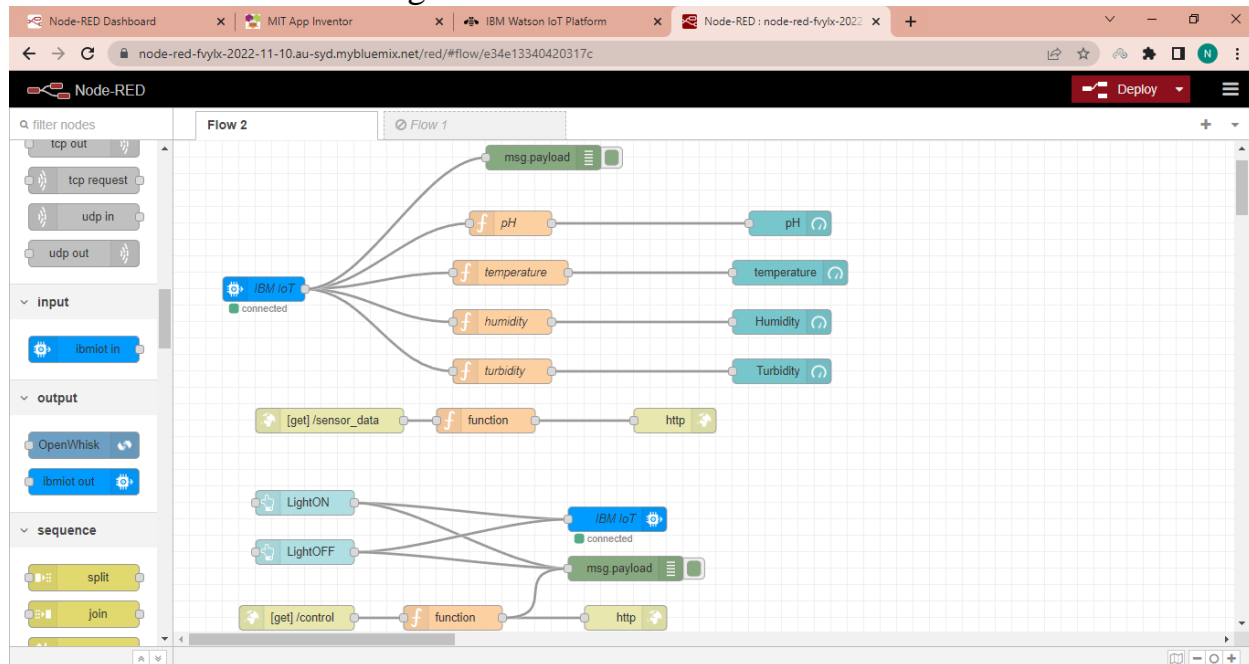
```
msg.payload: Object
  pH: 6, temperature: 18, turbidity: 62, humidity: 71
11/13/2022, 9:46:58 PM node: 1d80527787dc9cb7
iot-2/type/nodemcu/2002/ev/evnt_1/fmt/json :
msg.payload: Object
  pH: 4, temperature: 45, turbidity: 33, humidity: 50
11/13/2022, 9:47:03 PM node: 1d80527787dc9cb7
iot-2/type/nodemcu/2002/ev/evnt_1/fmt/json :
msg.payload: Object
  pH: 1, temperature: 14, turbidity: 2, humidity: 3
11/13/2022, 9:47:07 PM node: 1d80527787dc9cb7
iot-2/type/nodemcu/2002/ev/evnt_1/fmt/json :
msg.payload: Object
  pH: 2, temperature: 65, turbidity: 12, humidity: 51
11/13/2022, 9:47:09 PM node: 1d80527787dc9cb7
iot-2/type/nodemcu/2002/ev/evnt_1/fmt/json :
msg.payload: Object
  pH: 3, temperature: 78, turbidity: 22, humidity: 53
```

## STEP 9: Edit button mode [light ON and light OFF].

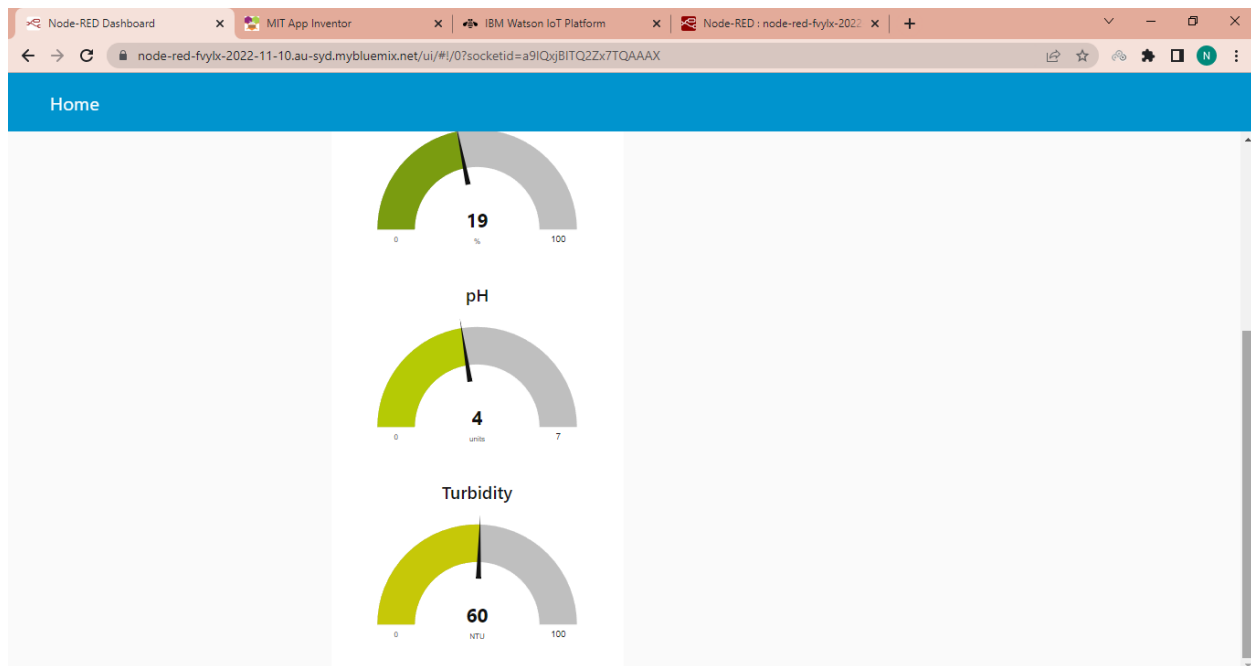
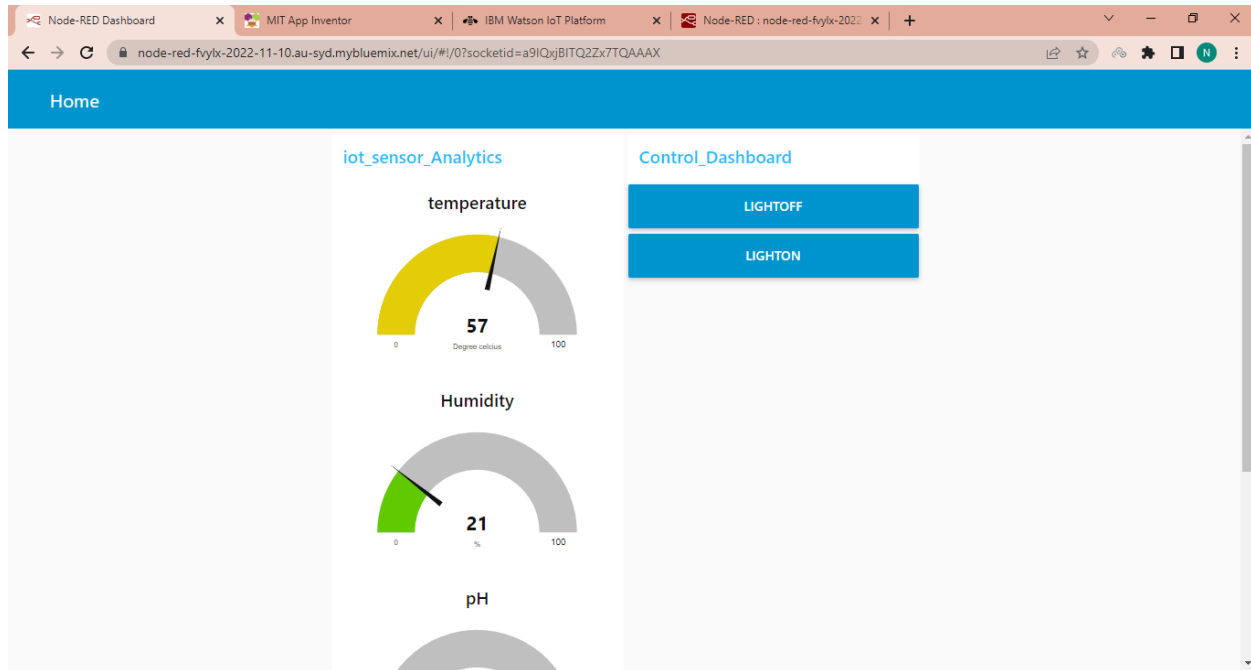
The screenshot shows the Node-RED web interface. The 'Edit button node' dialog is open for the 'LightON' button. The 'Group' is set to '[Home] Control\_Dashboard'. The 'Label' is 'LightON'. The 'Payload' is 'lighton' and the 'Topic' is 'msg.topic'. The 'When clicked, send:' section is visible. The 'debug' console on the right shows a series of messages with pH, temperature, and turbidity data.

```
msg.payload: Object
  pH: 4, temperature: 18, turbidity: 26, humidity: 9 }
11/13/2022, 9:48:24 PM node: 1d80527787dc9cb7
iot-2/type/nodemcu/2002/ev/1/event_1/fmt/json :
msg.payload: Object
  pH: 1, temperature: 2, turbidity: 79, humidity: 19 }
11/13/2022, 9:48:27 PM node: 1d80527787dc9cb7
iot-2/type/nodemcu/2002/ev/1/event_1/fmt/json :
msg.payload: Object
  pH: 3, temperature: 55, turbidity: 13, humidity: 55 }
11/13/2022, 9:48:31 PM node: 1d80527787dc9cb7
iot-2/type/nodemcu/2002/ev/1/event_1/fmt/json :
msg.payload: Object
  pH: 0, temperature: 67, turbidity: 63, humidity: 84 }
11/13/2022, 9:48:33 PM node: 1d80527787dc9cb7
iot-2/type/nodemcu/2002/ev/1/event_1/fmt/json :
msg.payload: Object
  pH: 0, temperature: 99, turbidity: 16, humidity: 18 }
```

## STEP 10: Entire flow diagram in Node-RED.

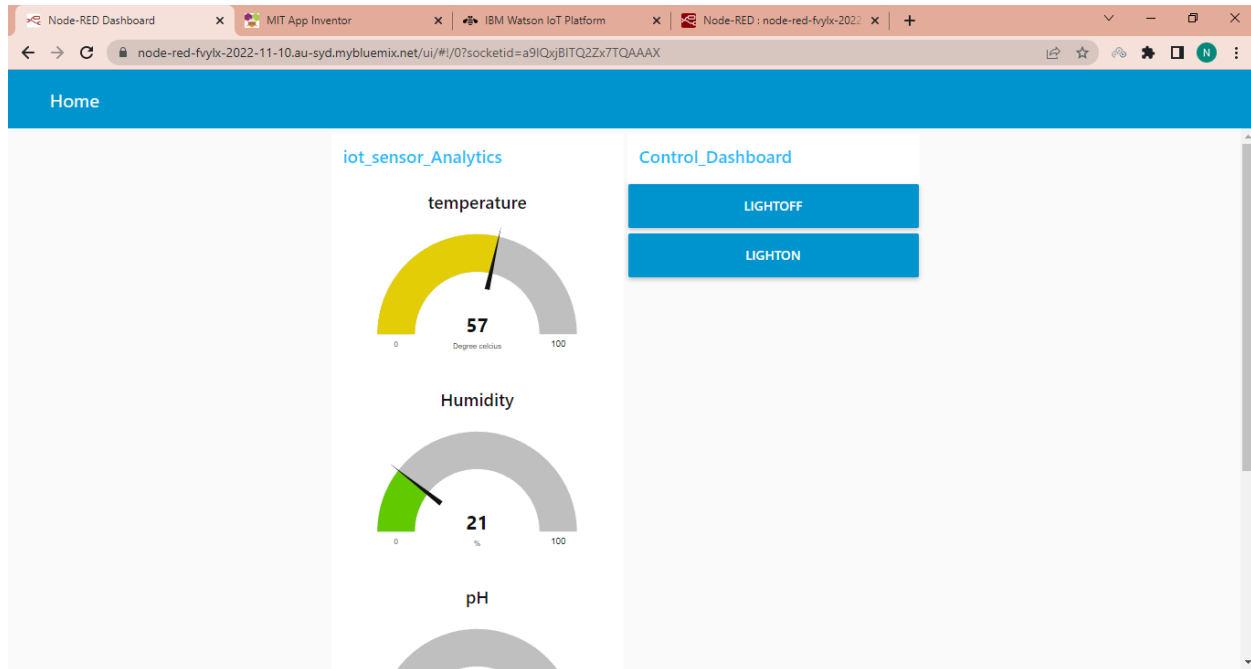


## STEP 11: Generate the output from recent events.





## STEP 12: Implementing url in the function node to generate output



The screenshot shows a web browser window with the same tabs as the previous image. The active tab is titled 'https://node-red-fylix-2022-'. The address bar shows the URL: `node-red-fylix-2022-11-10.au-syd.mybluemix.net/sensor_data`. The main content area displays a JSON object: `{"temperature":100,"humidity":72,"pH":4,"turbidity":46}`.

