

## Project development phase

### Sprint - 2

Date	15 November 2022
Team ID	PNT2022TMID26511
Project Name	Project - Industry-specific intelligent fire management system
Maximum Marks	20 marks

#### ▼ IN Sprint 2 31 Oct – 5 Nov (2 issues)

IN-4 In industry, sensor sense the fire and smoke. **SENSOR & ACTUATOR**

IN-5 If the sensor detected the fire, next step is extinguishing the fire with the help of Sprinkler. **SENSOR & ACTUATOR**

- ⇒ Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform.



Browse IBM Cloud Apps

## The API key has been added.

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.

### Generated Details

API Key a-dvo306-lmrrimazvy

Authentication Token 08+bzC68Sx0bZYDSCL



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.

### API Key Information

Description -

Role Standard Application

Expires Never

[View API Key](#)[Add Another](#)[Close](#)

## Browse API Keys

This table shows a summary of the API keys that have been added for the organization. It can be filtered.

1 Simulation running

27°C Cloudy

22:51

12/11/2022

## US-2 Create a Node-RED service

node-red-uwrpn-2022-11-12.eu-gb.mybluemix.net/red/#flow/caec888b429a5cb

Brown Purple Beige...

Node-RED : node-r...

IBM Watson IoT Pla...

IBM App Developm...

IBM Cloud Account

Node-RED

Deploy

filter nodes

Flow 1

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

function

switch

change

range

IBM IoT

connected

msg.payload

Temperature node

Temperature

Flame level

Flame level

Gas level node

Gas level

debug

all nodes

all

msg.payload : Object

{ Temperature: 60, Flame level: 30, Gas level: 37 }

11/13/2022, 7:08:36 PM node: aa0ccf3bbeaf648e iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json : msg.payload : Object

{ Temperature: 30, Flame level: 41, Gas level: 40 }

11/13/2022, 7:08:36 PM node: aa0ccf3bbeaf648e iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json : msg.payload : Object

{ Temperature: 65, Flame level: 89, Gas level: 77 }

11/13/2022, 7:08:36 PM node: aa0ccf3bbeaf648e iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json : msg.payload : Object

{ Temperature: 97, Flame level: 97, Gas level: 11 }

11/13/2022, 7:08:41 PM node: aa0ccf3bbeaf648e iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json : msg.payload : Object

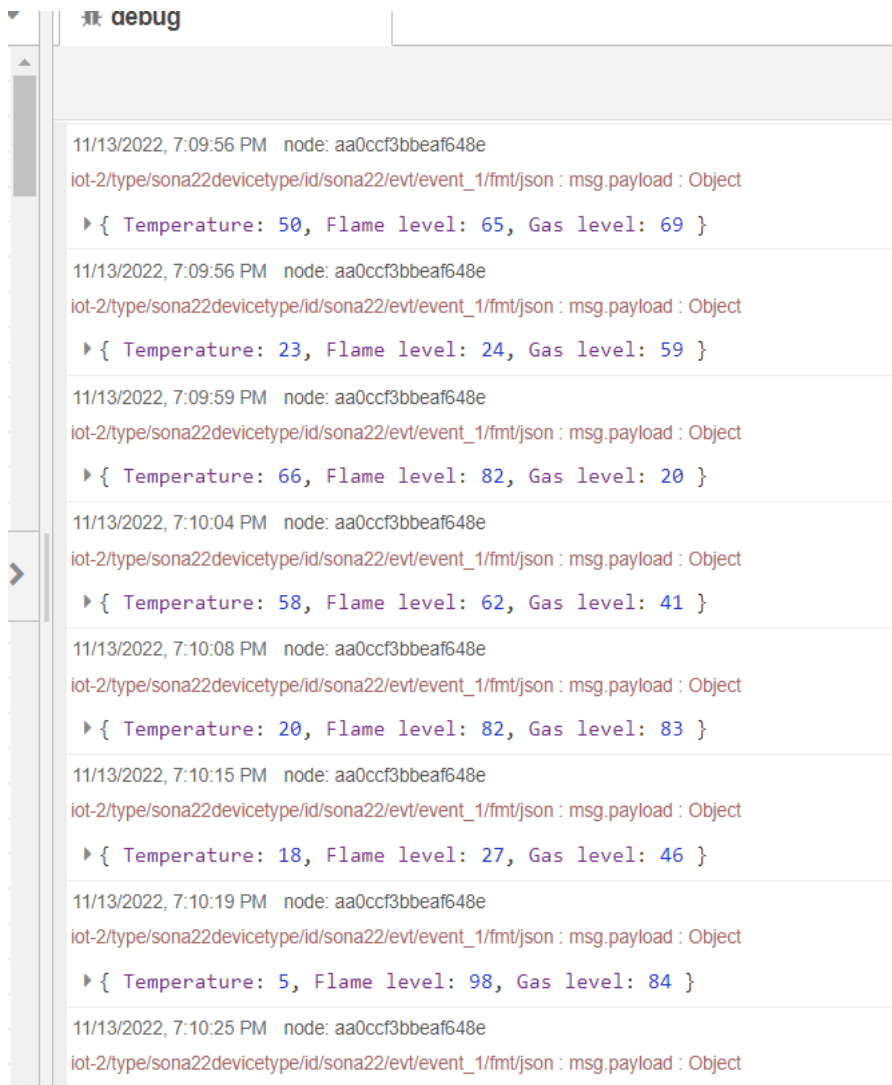
{ Temperature: 49, Flame level: 81, Gas level: 50 }

Type here to search

26°C Mostly cloudy

19:08 13-11-2022

**Fig1** - Monitoring the sensor values - Temperature, Flame Level, Gas Level. These values are randomly generated by IBM WATSON IOT PLATFORM.

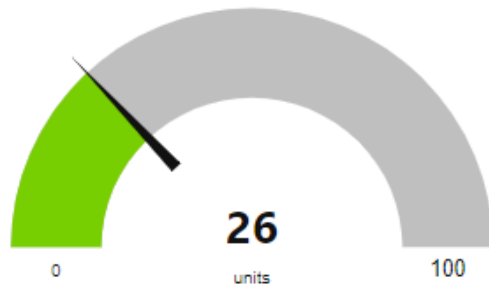


The screenshot displays the debug console of the IBM Watson IoT Platform. It shows a series of log entries for a device with node ID 'aa0ccf3bbeaf648e'. Each entry represents a sensor data update at a specific time. The logs include the timestamp, the node ID, the MQTT topic 'iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json', and the message payload which is a JSON object containing Temperature, Flame level, and Gas level values.

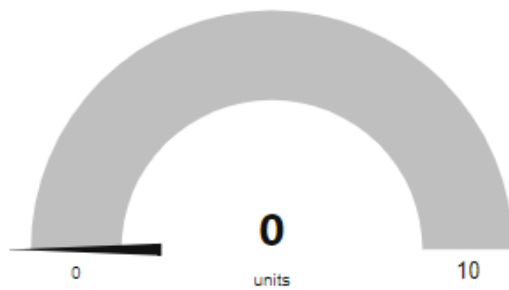
Timestamp	Node ID	Topic	Payload (JSON)
11/13/2022, 7:09:56 PM	aa0ccf3bbeaf648e	iot-2/type/sona22devicetype/id/sona22/evt/event_1/fmt/json	{ Temperature: 50, Flame level: 65, Gas level: 69 }
11/13/2022, 7:09:56 PM	aa0ccf3bbeaf648e	iot-2/type/sona22devicetype/id/sona22/evt/event_1/fmt/json	{ Temperature: 23, Flame level: 24, Gas level: 59 }
11/13/2022, 7:09:59 PM	aa0ccf3bbeaf648e	iot-2/type/sona22devicetype/id/sona22/evt/event_1/fmt/json	{ Temperature: 66, Flame level: 82, Gas level: 20 }
11/13/2022, 7:10:04 PM	aa0ccf3bbeaf648e	iot-2/type/sona22devicetype/id/sona22/evt/event_1/fmt/json	{ Temperature: 58, Flame level: 62, Gas level: 41 }
11/13/2022, 7:10:08 PM	aa0ccf3bbeaf648e	iot-2/type/sona22devicetype/id/sona22/evt/event_1/fmt/json	{ Temperature: 20, Flame level: 82, Gas level: 83 }
11/13/2022, 7:10:15 PM	aa0ccf3bbeaf648e	iot-2/type/sona22devicetype/id/sona22/evt/event_1/fmt/json	{ Temperature: 18, Flame level: 27, Gas level: 46 }
11/13/2022, 7:10:19 PM	aa0ccf3bbeaf648e	iot-2/type/sona22devicetype/id/sona22/evt/event_1/fmt/json	{ Temperature: 5, Flame level: 98, Gas level: 84 }
11/13/2022, 7:10:25 PM	aa0ccf3bbeaf648e	iot-2/type/sona22devicetype/id/sona22/evt/event_1/fmt/json	{ Temperature: 5, Flame level: 98, Gas level: 84 }

## Weather Monitoring

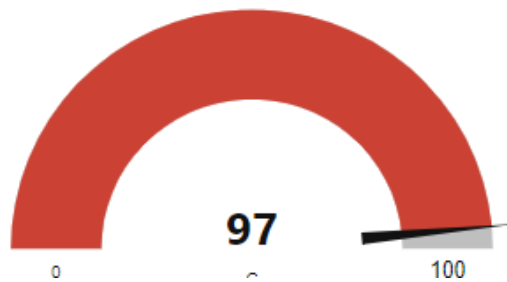
Flame level



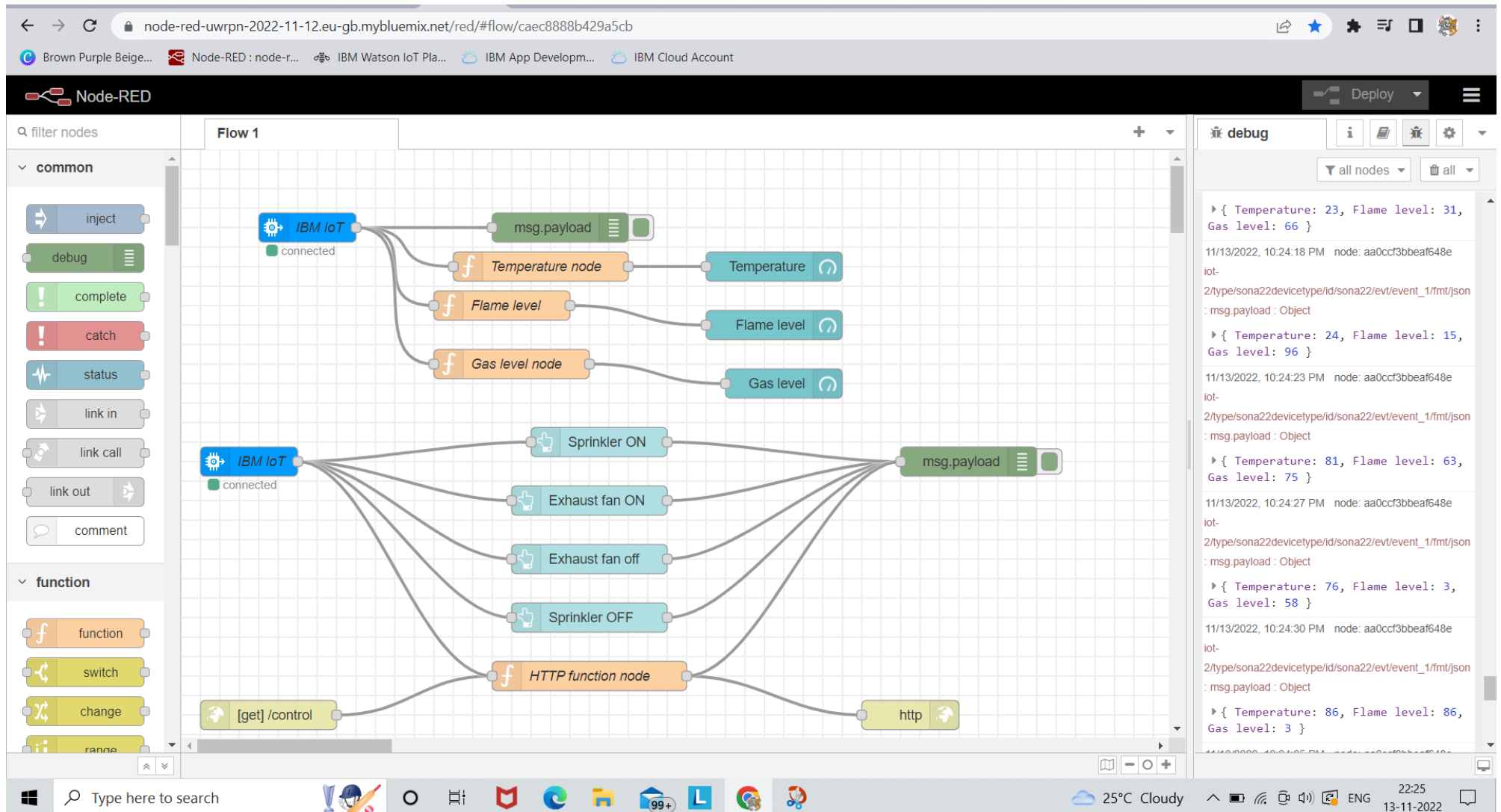
Gas level



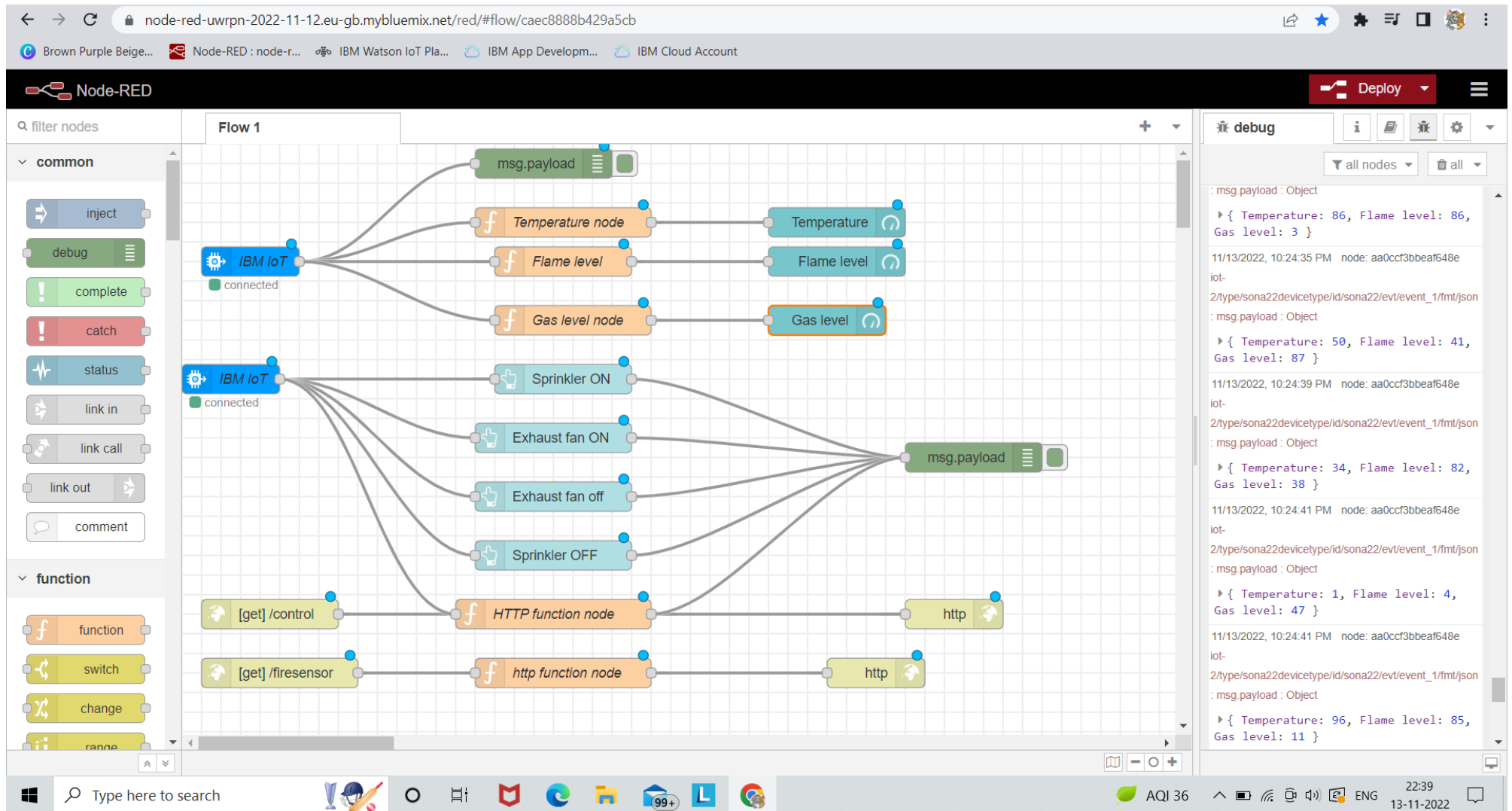
Temperature



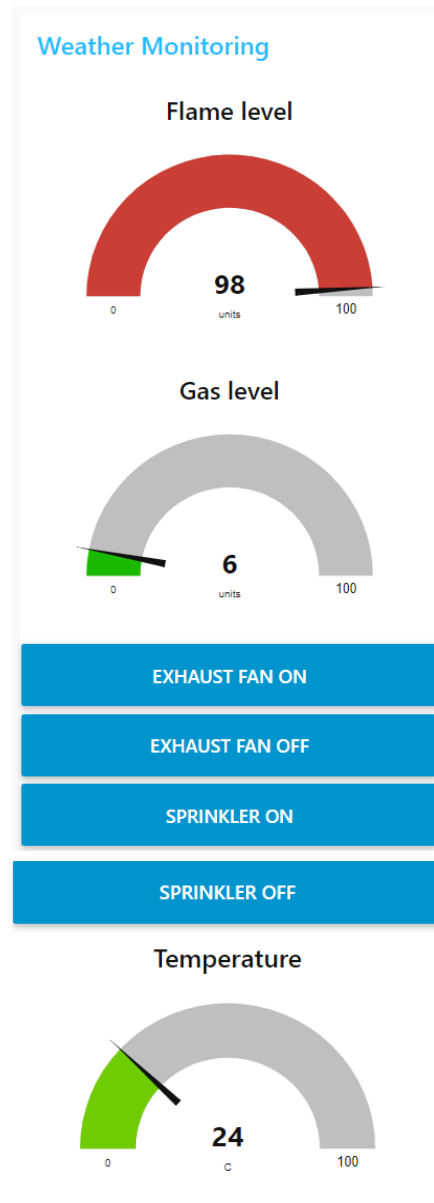
**Fig 2** - Temperature, Flame Level, Gas Level values displayed in deploy tab in node-red



**Fig 3 :** Using HTTP in and HTTP response in network option , <https://node-red-uwprn-2022-11-12.eu-gb.mybluemix.net/red/#flow/caec888b429a5cb> will display sensor values from the IBM WATSON IOT PLATFORM



**Fig 4 :** Monitoring the temperature ,flame and gas sensor





**Fig 5** : Properties of ibm iot node

Edit ibmiot in node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🔗

🔑 Authentication

API Key

▼

🔑 API Key

d709a7c4dcd74895

▼

✎

⚙ Input Type

Device Event

▼

🔑 Device Type

☐ All or

sona22devicetype

👤 Device Id

☐ All or

sona22

📋 Event

☒ All or

+

📄 Format

☐ All or

json

⚙ QoS

0

▼

🏷 Name

IBM IoT

🏷 Service

registered

Use the Input Type property to configure this node to receive Events sent by IoT Devices. Commands sent to IoT Devices. Status

☐ Enabled

**Fig 6 :** Properties of temperature node

Edit gauge node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🔍

🗃 Group

[control] Weather Monitoring

▼

✎

📏 Size

auto

☰ Type

Gauge

▼

🏷 Label

Temperature

🏷 Value format

{{value}}

🏷 Units

C

Range

min

0

max

100

Colour gradient

Sectors

0

...

optional

...

optional

...

100

🏷 Name

☐ Enabled

Fig 7: Properties of Flame

**Edit gauge node**

Delete Cancel Done

**Properties** [Settings] [Help] [Preview]

**Group** [control] Weather Monitoring [Edit]

**Size** auto

**Type** Gauge [v]

**Label** Flame level

**Value format** {{value}}

**Units** units

**Range** min 0 max 100

**Colour gradient** [Green] [Yellow] [Red]

**Sectors** 0 ... optional ... optional ... 100

**Name** [ ]

Fig 8 : Properties of Gas level node

Edit gauge node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

📁 Group

[control] Weather Monitoring

▼

✎

📏 Size

auto

☰ Type

Gauge

▼

🏷 Label

Gas level

🏷 Value format

{{value}}

🏷 Units

units

Range

min

0

max

10

Colour gradient

Sectors

0

...

optional

...

optional

...

10

🏷 Name

☐ Enabled

Fig 9 - Properties of IBM IOT are shown. The API key, Device Type, Device ID are taken from IBM IOT WATSON PLATFORM.

## Edit function node

Delete

Cancel

Done

### ⚙ Properties



📌 Name

Temperature node



⚙ Setup

On Start

**On Message**

On Stop

```
1 msg.payload=msg.payload.Temperature
2 global.set("t",msg.payload)
3 return msg;
```

Edit function node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

📁 Name

Flame level

📄

⚙ Setup

On Start

On Message

On Stop

1 msg.payload=msg.payload["Flame level"]

2 global.set("f",msg.payload)

3 return msg;

☐ Enabled

Edit function node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

📁 Name

Gas level node

📄

⚙ Setup

On Start

On Message

On Stop

1 msg.payload=msg.payload["Gas level"]

2 global.set("g",msg.payload)

3 return msg;

☐ Enabled

**Fig 10** - Properties of HTTP request with method GET and url control

The image shows a software interface for editing an HTTP node. The title bar reads "Edit http in node". Below the title bar are three buttons: "Delete", "Cancel", and "Done". The "Done" button is highlighted in red. The main area is titled "Properties" and contains three fields: "Method" with a dropdown menu set to "GET", "URL" with a text input containing "/control", and "Name" with a text input containing "Name". At the bottom left, there is a checkbox labeled "Enabled" which is checked.

**Edit http in node**

Delete Cancel Done

**Properties**

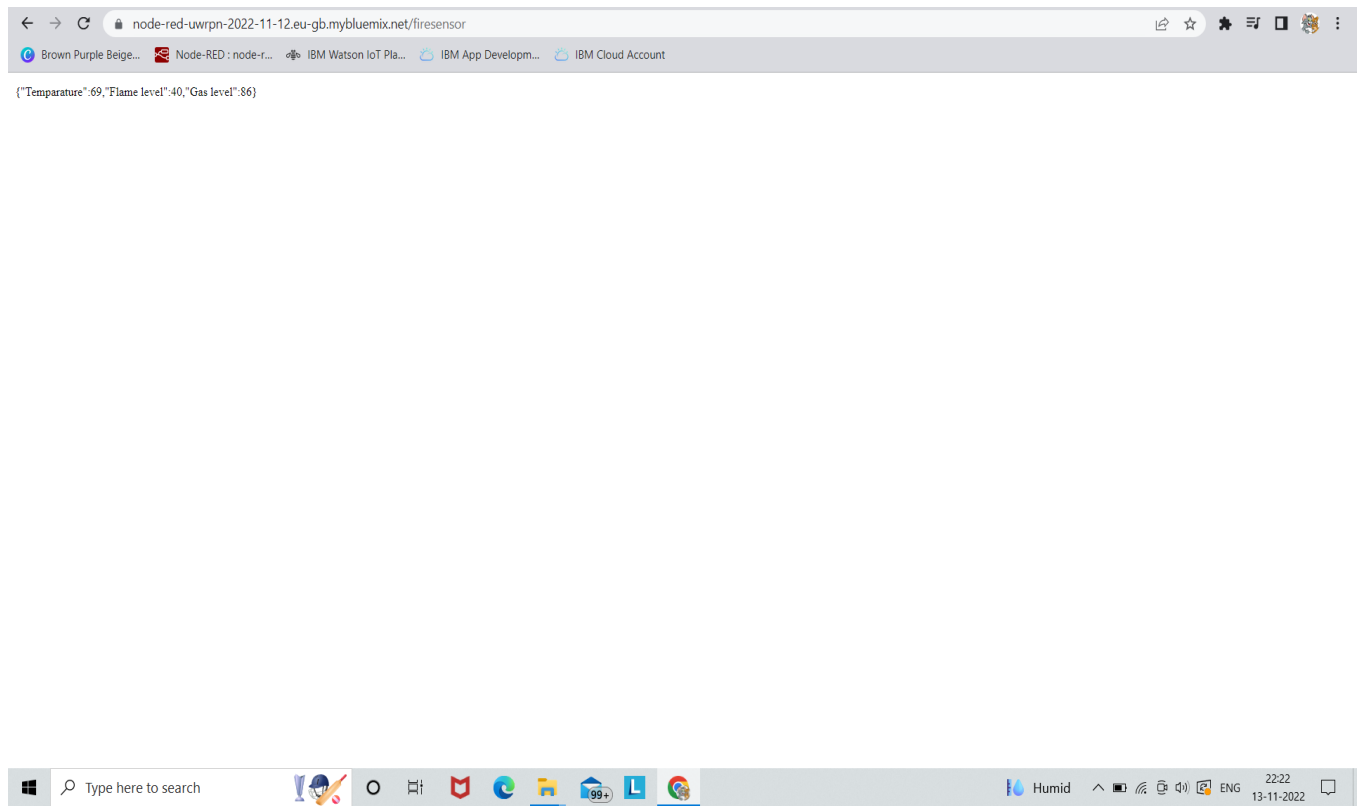
Method GET

URL /control

Name Name

Enabled

**Fig 11 : HTTP request OUTPUT**

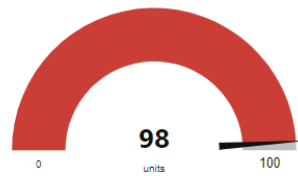




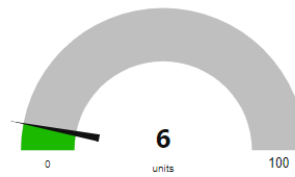
## control

### Weather Monitoring

#### Flame level



#### Gas level



EXHAUST FAN ON

EXHAUST FAN OFF

SPRINKLER ON

**FIG 12 :** Front -end APP for our project to display the temperature ,smoke level and flame level with control buttons like Sprinkler ,exhaust fan on and off buttons

