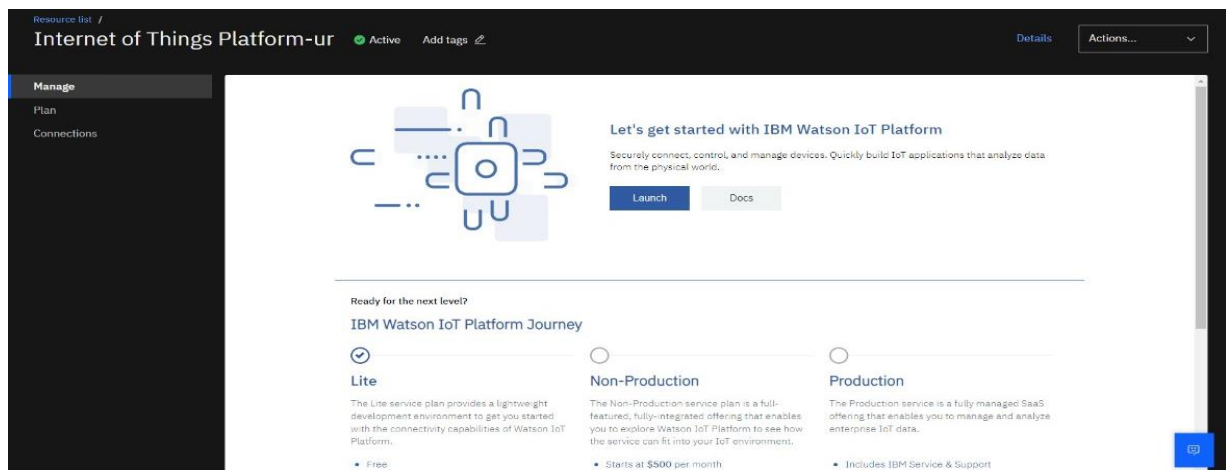


Sprint 2

Date	03 November 2022
Team ID	PNT2022TMID31297
Project Name	Smart Farmer-IoT Enabled smart Farming Application
Maximum Marks	4 Marks

IBM Watson IoT Platform



Steps to configure:

- Create an account in IBM cloud using your email ID
- Create IBM Watson Platform in services in your IBM cloud account
- Launch the IBM Watson IoT Platform
- Create a new device
- Give credentials like device type, device ID, Auth. Token
- Create API key and store API key and token elsewhere.

Create a new device:

In our project in the place of sensors we are going to use IoT sensor simulator which give random readings to the connected cloud.

Connecting IoT Simulator to IBM Watson IoT Platform:

My credentials given to simulator are:

Org:1xl08d

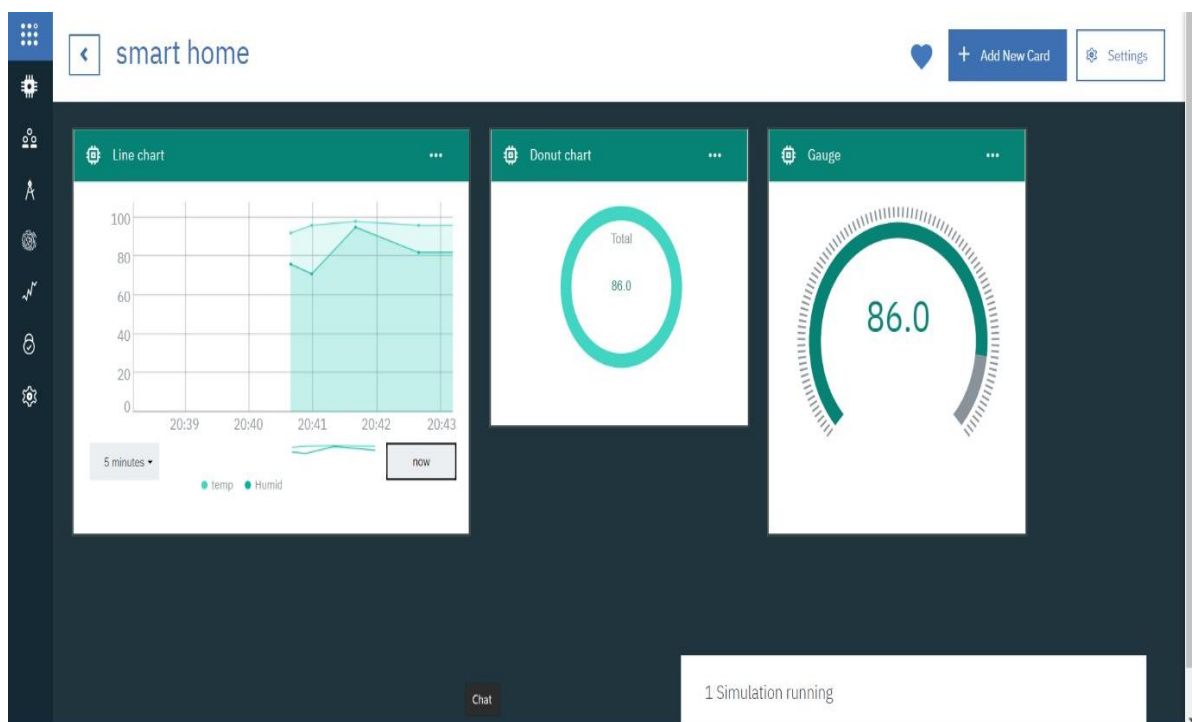
API: a-1xl08d-p5eywn2eu

Auth Token:GpIJ5spsrx0ZB*RLmJ

Device Type:abcd

Device ID:12

Device Token:12345678



You can see the received data in graphs by creating cards in Boards tab

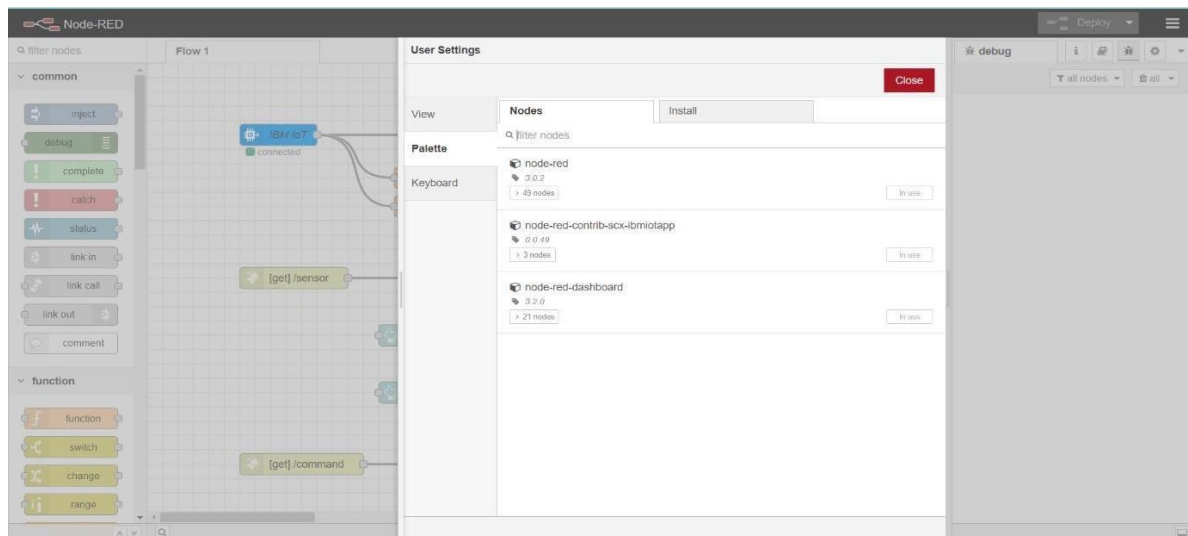
- You will receive the simulator data in cloud
- You can see the received data in Recent Events under your device
- Data received in this format(json)

Configuring IBM-IoT to Node-RED connection:

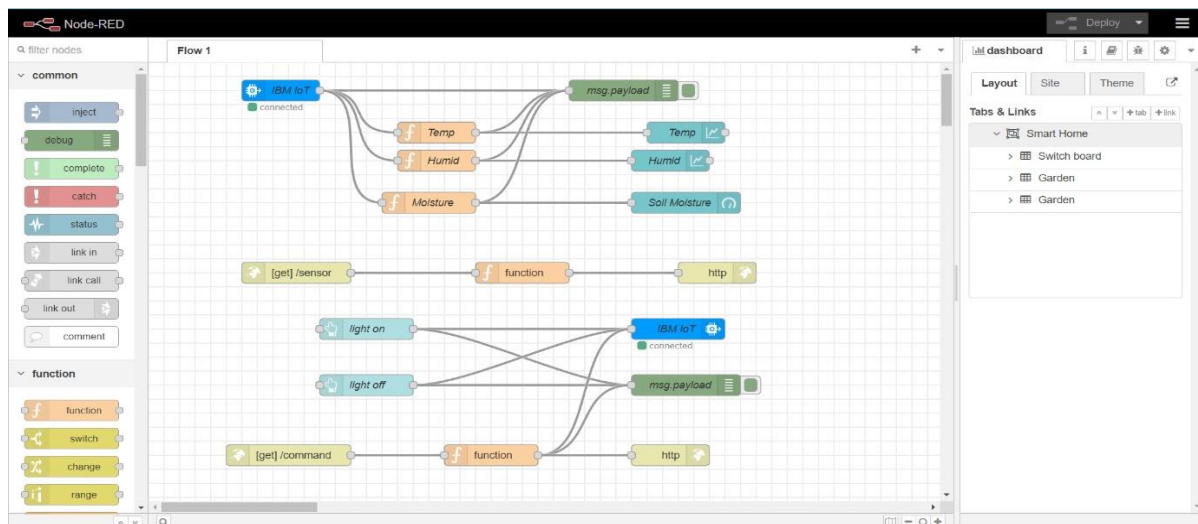
The screenshot displays the IBM IoT Dashboard interface. On the left, a sidebar contains navigation icons. The main panel shows a table of devices with columns: Device ID, Status, Device Type, Class ID, and Date. A device with ID '12' and status 'Disconnected' is selected. Below the table, the 'Recent Events' tab is active, showing a list of events with columns: Event, Value, Format, and Last Received. The events are in JSON format, containing fields like 'randomNumber', 'temp', and 'humid'. On the right, a modal window titled 'Device Type: abcd' is open, showing the 'Events' configuration. It includes a 'New event type' button, a 'Send' button, and a 'Schedule' section set to 'Every Minute'. The 'Payload' section shows a JSON object with random values for 'temp' and 'humid'.

The screenshot shows the Node-RED interface. On the left, a sidebar lists various nodes under 'common' and 'function' categories. The main workspace displays a flow with several nodes: 'inject', 'debug', 'complete', 'catch', 'status', 'link in', 'link call', 'link out', 'comment', 'function', 'switch', 'change', and 'range'. The 'IBM IoT' node is highlighted. On the right, the 'Edit IBM IoT node' configuration panel is open, showing properties such as Name (API), API Key (a-1x08d-p5eywn2eu), API Token (*****), Server-Name (orgid.messaging.internetofthings.ibmcloud.com), Scalable (checkbox), Application ID, Keep Alive (60 Seconds), and Use Clean Session (checkbox). The 'Update' button is visible.

Installing a node-red-contrib-scx-ibmiotapp and node-red dashboard

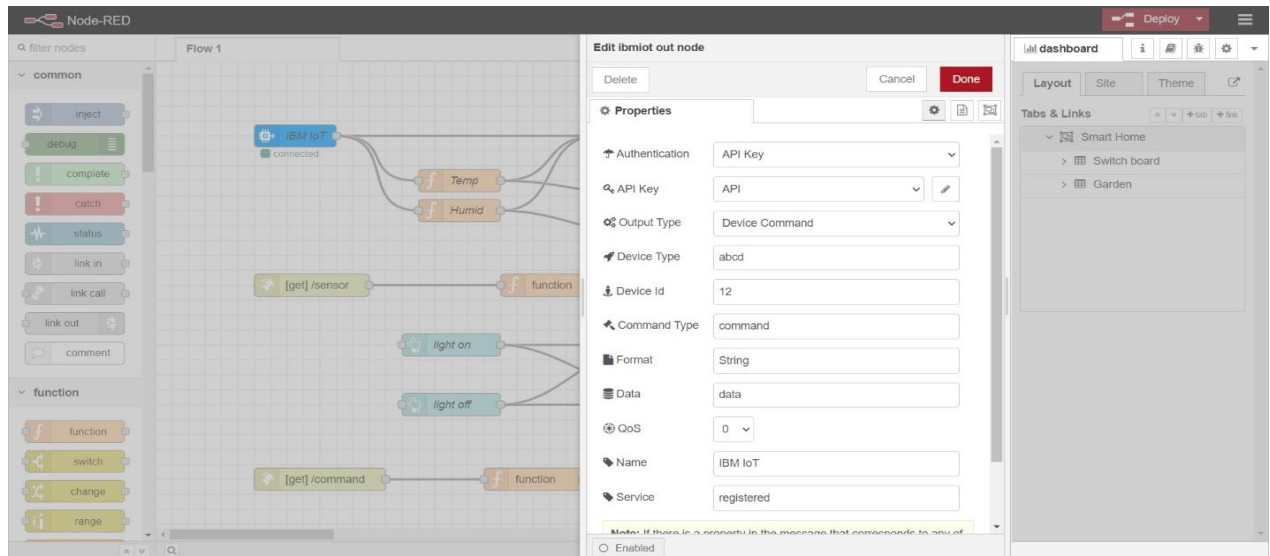


Complete Program Flow:



Configuration of Node-Red to collect IBM cloud data:

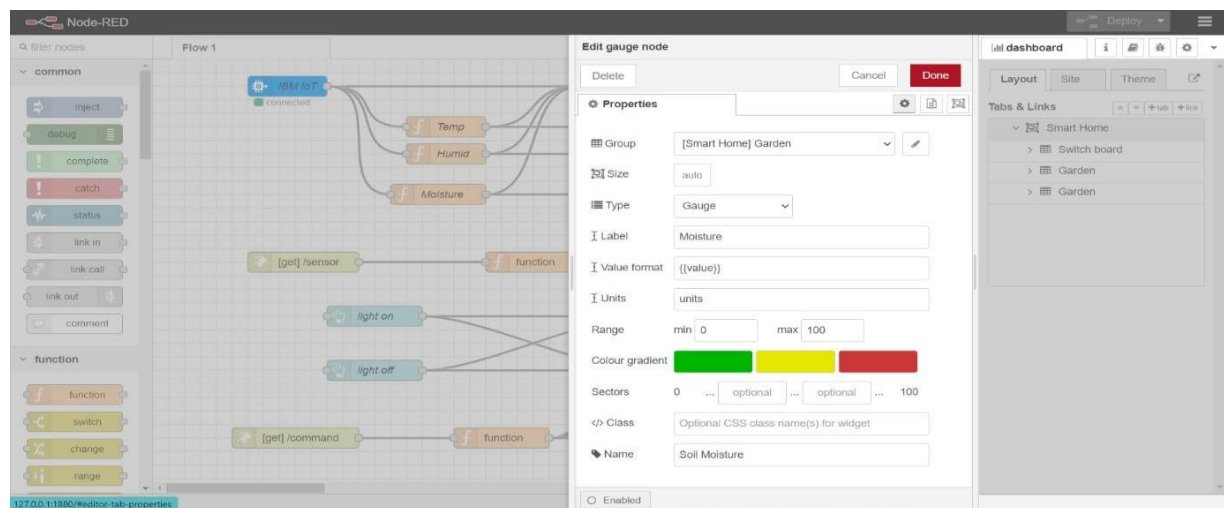
The node IBM IoT App In is added to Node-Red workflow. Then the appropriate device credentials obtained earlier are entered into the node to connect and fetch device telemetry to Node-Red.



Connect function node and The Java Script code for the function node is:

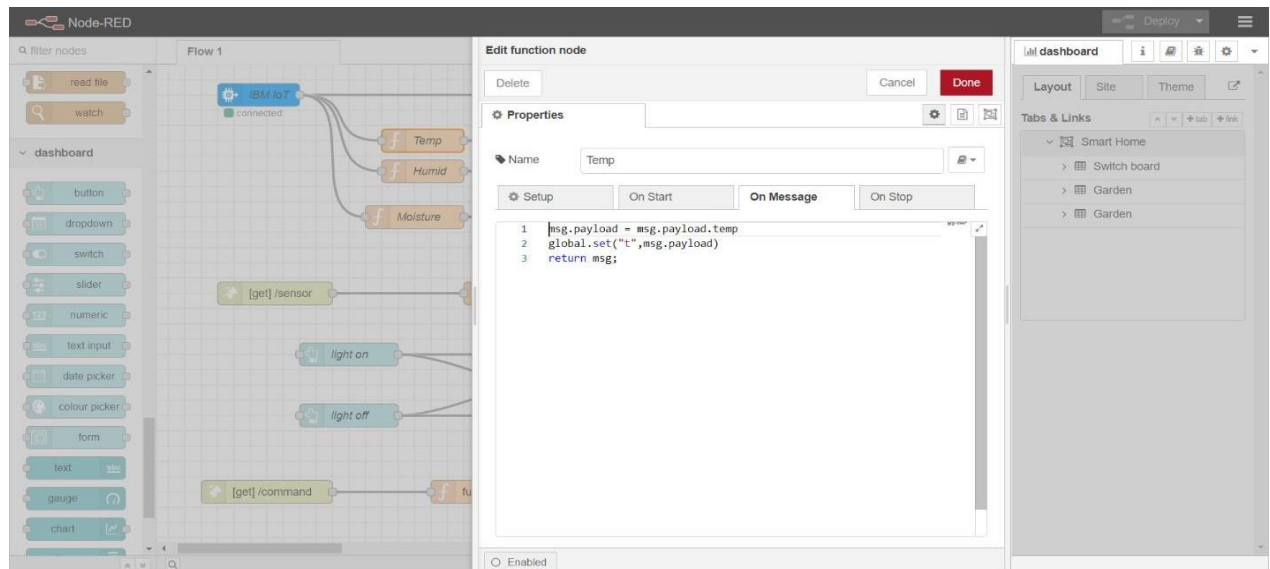
```
msg.payload=msg.payload.temp
return msg;
```

Finally connect Gauge nodes from dashboard to see the data in UI

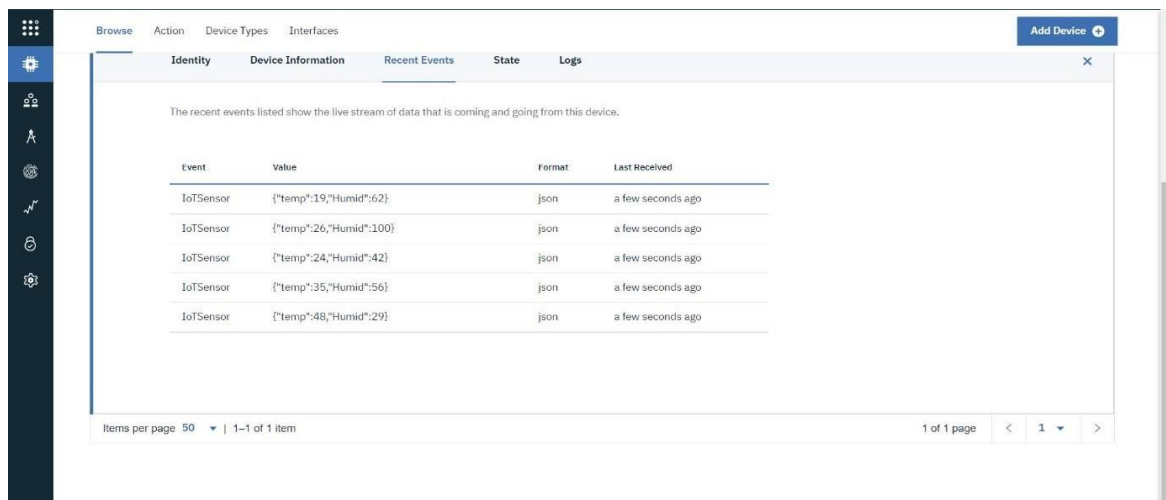


Configuration of Node-Red to collect data from OpenWeather:

The Node-Red also receive data from the OpenWeather API by HTTP GET request. An inject trigger is added to perform HTTP request for every certain interval.



Checking IoT sensor Output in IBM Watson:



Checking IoT sensor using command in Node-RED:

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
=== RESTART: C:\Users\Sneha\AppData\Local\Programs\Python\Python37\sum.py ===
2022-11-04 21:25:30.628 ibmiotf.device.Client INFO Connected successfully: dr1x105d:abcd:12
Published Temperature = 64 C Humidity = 32 % to IBM Watson
Published Temperature = 50 C Humidity = 5 % to IBM Watson
Published Temperature = 50 C Humidity = 61 % to IBM Watson
Published Temperature = 34 C Humidity = 32 % to IBM Watson
Published Temperature = 32 C Humidity = 87 % to IBM Watson
Command received: lighton
led is on
Published Temperature = 87 C Humidity = 41 % to IBM Watson
Published Temperature = 5 C Humidity = 61 % to IBM Watson
Published Temperature = 28 C Humidity = 0 % to IBM Watson
Published Temperature = 73 C Humidity = 12 % to IBM Watson
Published Temperature = 62 C Humidity = 26 % to IBM Watson
Published Temperature = 77 C Humidity = 62 % to IBM Watson
Published Temperature = 25 C Humidity = 13 % to IBM Watson
Published Temperature = 57 C Humidity = 100 % to IBM Watson
Published Temperature = 42 C Humidity = 77 % to IBM Watson
Command received: lightoff
led is off
Published Temperature = 76 C Humidity = 13 % to IBM Watson
Published Temperature = 57 C Humidity = 21 % to IBM Watson
Published Temperature = 55 C Humidity = 57 % to IBM Watson
Published Temperature = 13 C Humidity = 47 % to IBM Watson
Published Temperature = 75 C Humidity = 60 % to IBM Watson
Published Temperature = 51 C Humidity = 73 % to IBM Watson
Published Temperature = 71 C Humidity = 34 % to IBM Watson
```

Output in Node-RED Dashboard:

