

# WEB APPLICATION

**Step 1:** Generate random values of Humidity, Temperature, Soil Moisture are generated from events in the Watson IOT platform.

The screenshot displays the IBM Watson IoT Platform web application. The main page is titled 'Browse Devices' and shows a table with columns for Device ID, Status, and Device Type. A modal window is open for creating a new event type named 'event\_1'. The modal includes a 'Schedule' section set to 'Every Minute' and a 'Payload' section with a JSON payload containing random values for 'randomNumber', 'temp', and 'humidity'.

IBM Watson IoT Platform

Browse Action Device Types Interfaces

## Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered by criteria. To get started, you can add devices by using the Add Device button, or by using the Add Device button.

Search by Device ID

Device ID	Status	Device Type
12	Disconnected	abcd

Items per page 50 | 1-1 of 1 item

1 of 1 page

Device Type: abcd

Events 1

New event type +

Event type name event\_1 Send

Schedule

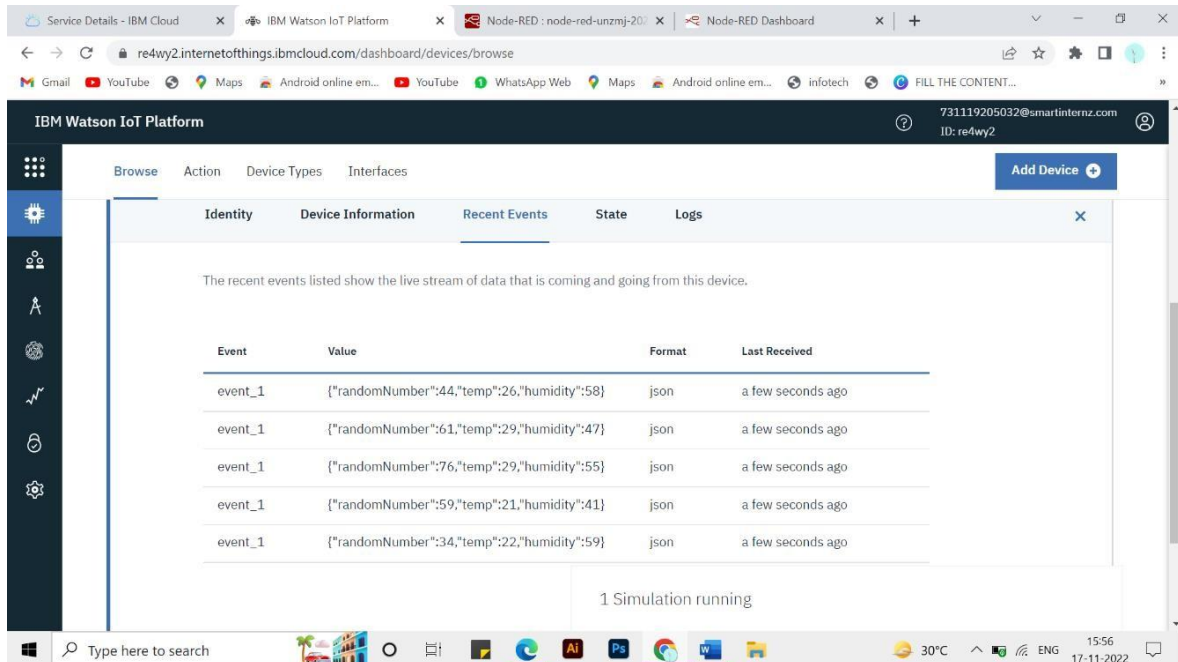
1 Every Minute

Payload

Specify the event payload in the editor window or by uploading a CSV file.

```
0 {
1   "randomNumber": random(21, 80),
2   "temp": random(20, 40),
3   "humidity": random(40, 60)
4 }
5 }
```

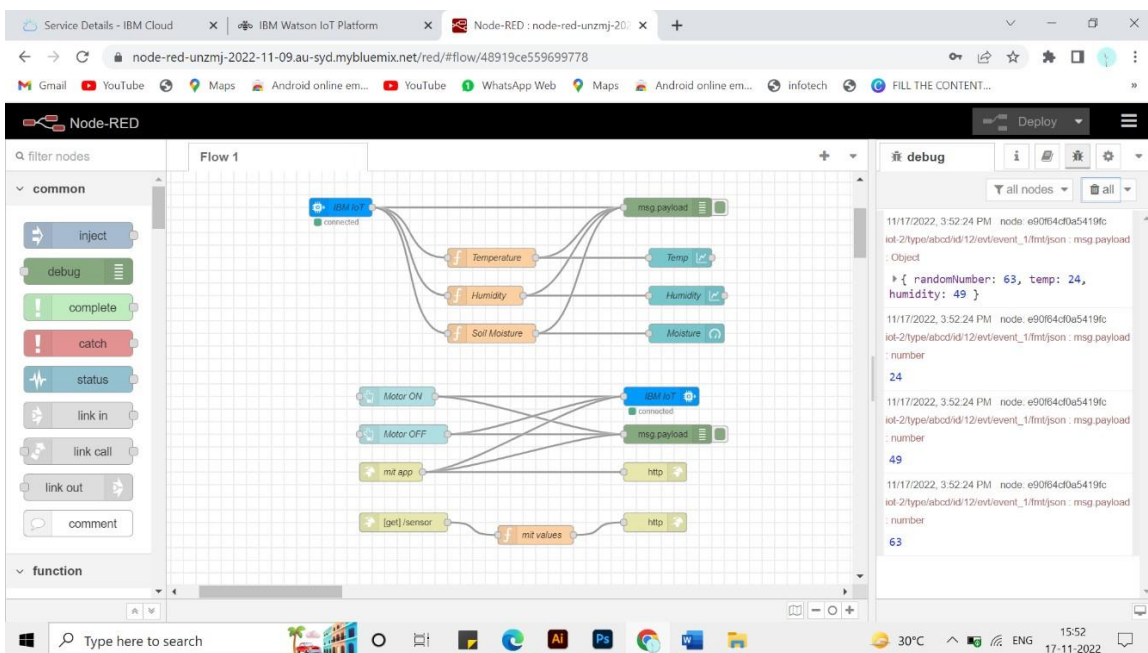
**Step 2:** The values are generated for every minute as payload from events in the form of json format.



The screenshot shows the IBM Watson IoT Platform dashboard. The 'Recent Events' tab is selected, displaying a table of events. The table has four columns: Event, Value, Format, and Last Received. The events are generated every minute and contain random data for temperature, humidity, and soil moisture. Below the table, it indicates '1 Simulation running'.

Event	Value	Format	Last Received
event_1	{"randomNumber":44,"temp":26,"humidity":58}	json	a few seconds ago
event_1	{"randomNumber":61,"temp":29,"humidity":47}	json	a few seconds ago
event_1	{"randomNumber":76,"temp":29,"humidity":55}	json	a few seconds ago
event_1	{"randomNumber":59,"temp":21,"humidity":41}	json	a few seconds ago
event_1	{"randomNumber":34,"temp":22,"humidity":59}	json	a few seconds ago

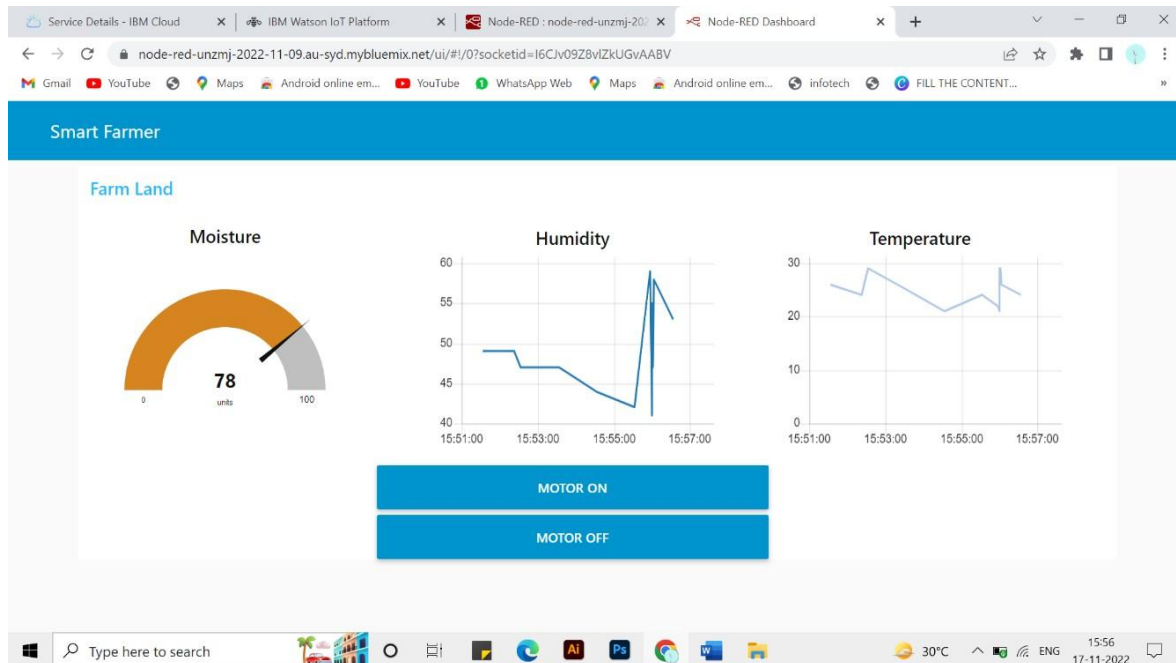
**Step 3:** Node-RED is an editor used to create the flow between the nodes and has to be deployed once the flow has been made. Once deployment is done the editor displays the details of temperature, soil moisture, humidity in the debug section.



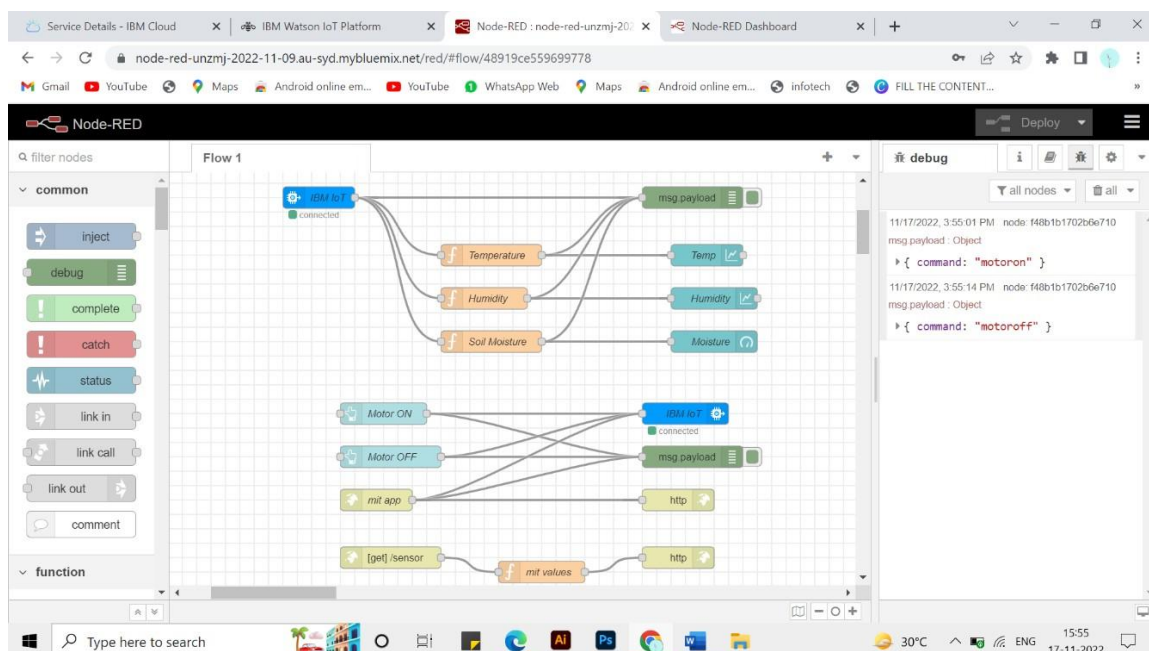
The screenshot shows the Node-RED editor interface. A flow named 'Flow 1' is visible, which processes incoming IoT data. The flow starts with an 'inject' node, followed by a 'msg.payload' node. The data is then split into three parallel paths: 'Temperature', 'Humidity', and 'Soil Moisture'. Each path uses a 'msg.payload' node to extract the specific value. These values are then sent to a 'debug' node for monitoring. The 'debug' node is configured to show the 'msg.payload' property. The flow is deployed, and the debug console shows the following output:

```
11/17/2022, 3:52:24 PM node: e90f64cd0a5419fc  
iot-2/type/abcd/id/12/event_1/fmt/json : msg.payload  
: Object  
  { randomNumber: 63, temp: 24,  
    humidity: 49 }  
11/17/2022, 3:52:24 PM node: e90f64cd0a5419fc  
iot-2/type/abcd/id/12/event_1/fmt/json : msg.payload  
: number  
  24  
11/17/2022, 3:52:24 PM node: e90f64cd0a5419fc  
iot-2/type/abcd/id/12/event_1/fmt/json : msg.payload  
: number  
  49  
11/17/2022, 3:52:24 PM node: e90f64cd0a5419fc  
iot-2/type/abcd/id/12/event_1/fmt/json : msg.payload  
: number  
  63
```

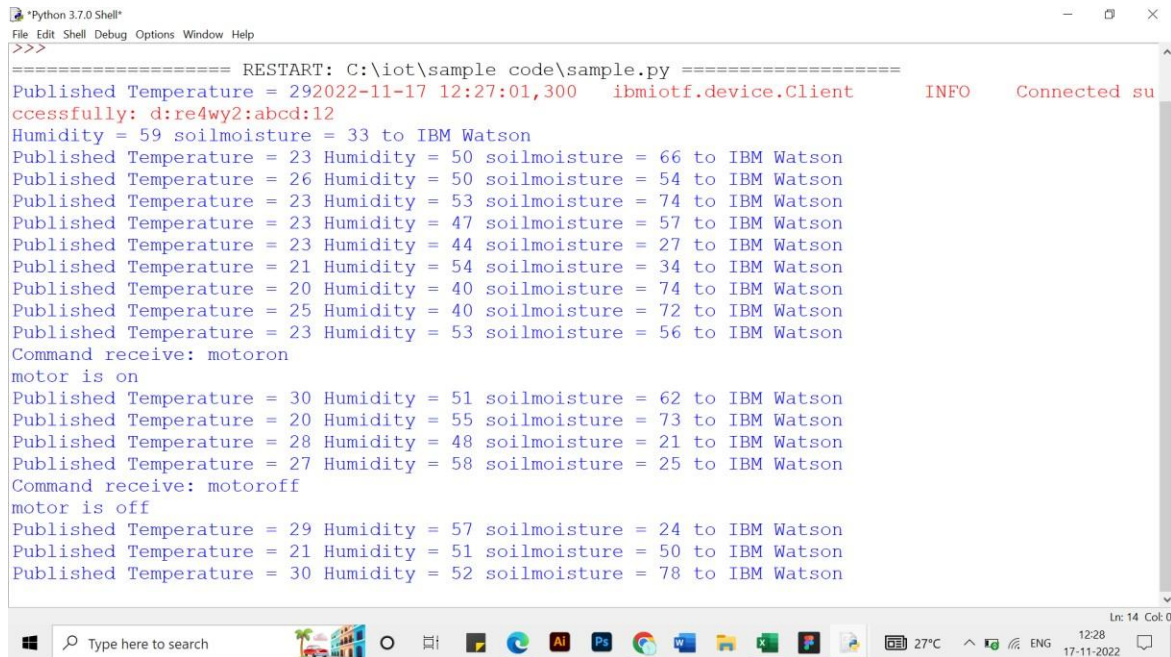
**Step 4:** The **Smart Farmer** dashboard is viewed once the deployment is completed where we can able to view the graphs of Humidity and Temperature. Gauge is used to view the moisture.



**Step 5:** When the **Motor ON** button is clicked we receive the output as “**motoron**” and **Motor OFF** button is clicked we receive the output as “**motoroff**”. And these output are received in the debug section of the editor.



**Step 6:** The output is also received in the **python code editor** when the buttons are clicked in the dashboard and random values are also generated.



```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
>>>
===== RESTART: C:\iot\sample code\sample.py =====
Published Temperature = 29 2022-11-17 12:27:01,300 ibmiotf.device.Client INFO Connected successfully: d:re4wy2:abcd:12
Humidity = 59 soilmoisture = 33 to IBM Watson
Published Temperature = 23 Humidity = 50 soilmoisture = 66 to IBM Watson
Published Temperature = 26 Humidity = 50 soilmoisture = 54 to IBM Watson
Published Temperature = 23 Humidity = 53 soilmoisture = 74 to IBM Watson
Published Temperature = 23 Humidity = 47 soilmoisture = 57 to IBM Watson
Published Temperature = 23 Humidity = 44 soilmoisture = 27 to IBM Watson
Published Temperature = 21 Humidity = 54 soilmoisture = 34 to IBM Watson
Published Temperature = 20 Humidity = 40 soilmoisture = 74 to IBM Watson
Published Temperature = 25 Humidity = 40 soilmoisture = 72 to IBM Watson
Published Temperature = 23 Humidity = 53 soilmoisture = 56 to IBM Watson
Command receive: motoron
motor is on
Published Temperature = 30 Humidity = 51 soilmoisture = 62 to IBM Watson
Published Temperature = 20 Humidity = 55 soilmoisture = 73 to IBM Watson
Published Temperature = 28 Humidity = 48 soilmoisture = 21 to IBM Watson
Published Temperature = 27 Humidity = 58 soilmoisture = 25 to IBM Watson
Command receive: motoroff
motor is off
Published Temperature = 29 Humidity = 57 soilmoisture = 24 to IBM Watson
Published Temperature = 21 Humidity = 51 soilmoisture = 50 to IBM Watson
Published Temperature = 30 Humidity = 52 soilmoisture = 78 to IBM Watson
```