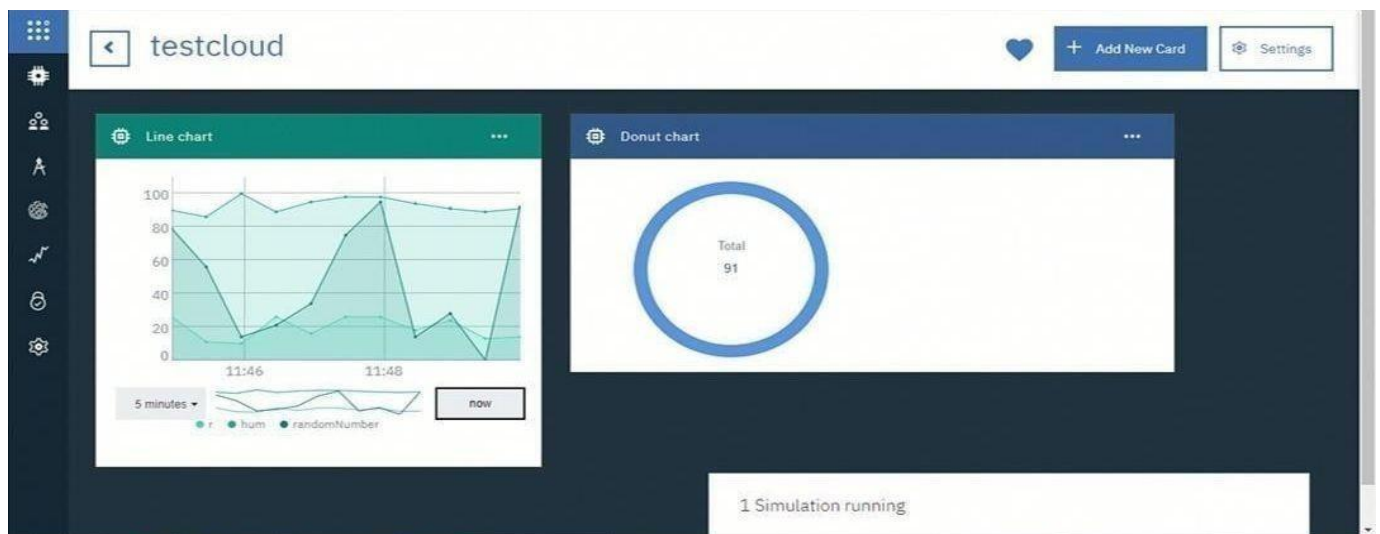


SPRINT DELIVERY – 2

TEAM ID	PNT2022TMID15961
PROJECT TITLE	SMART FARMER – IOT ENABLED SMART FARMING APPLICATION
TEAM LEADER	SANJAY G
TEAM MEMBER 1	PRAVEEN N
TEAM MEMBER 2	RAJESH N
TEAM MEMBER 3	RAM SRINIVAS C

Connecting IOT Simulator to IBM Watson



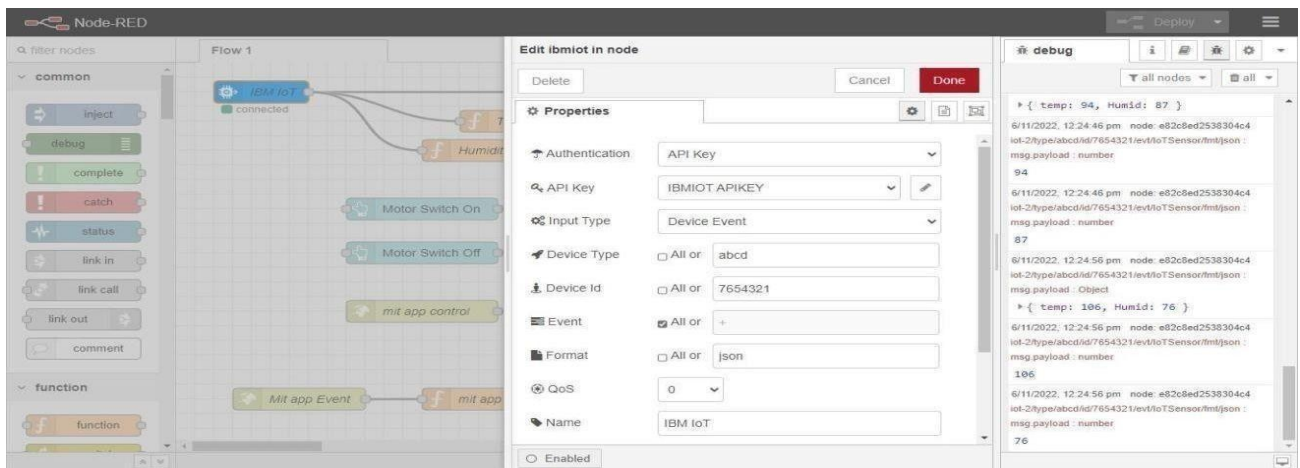
- You can see the received data in Recent Events under your device
- Data received in this format(json)

```
{  
  "d": {  
    ▪ "name": "NodeMCU",  
    ▪ "temperature": 17,  
    ▪ "humidity": 76,  
    ▪ "Moisture ": 25  
  }  
}
```

The screenshot shows a web interface for managing IoT devices. A sidebar on the left contains navigation icons. The main content area has tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, displaying a table of recent data received from a device. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. It shows three entries from an 'IoT Sensor' with JSON data containing temperature and humidity values. Below the table, there is a pagination control showing '1 of 1 page'.

Event	Value	Format	Last Received
IoT Sensor	{"temp":108,"Humid":64}	json	a few seconds ago
IoT Sensor	{"temp":91,"Humid":93}	json	a few seconds ago
IoT Sensor	{"temp":108,"Humid":83}	json	a few seconds ago

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.

Once it is connected Node-Red receives data from the device Display the data using debug node for verification

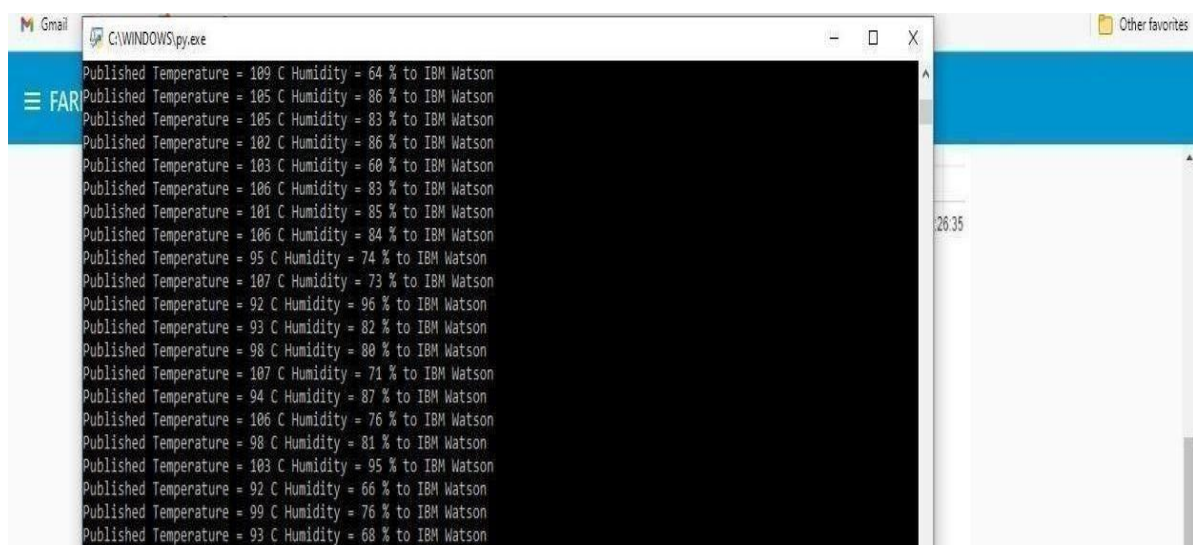
Connect function node and write the Java script code to get each reading separately.

The Java script code for the function node is:

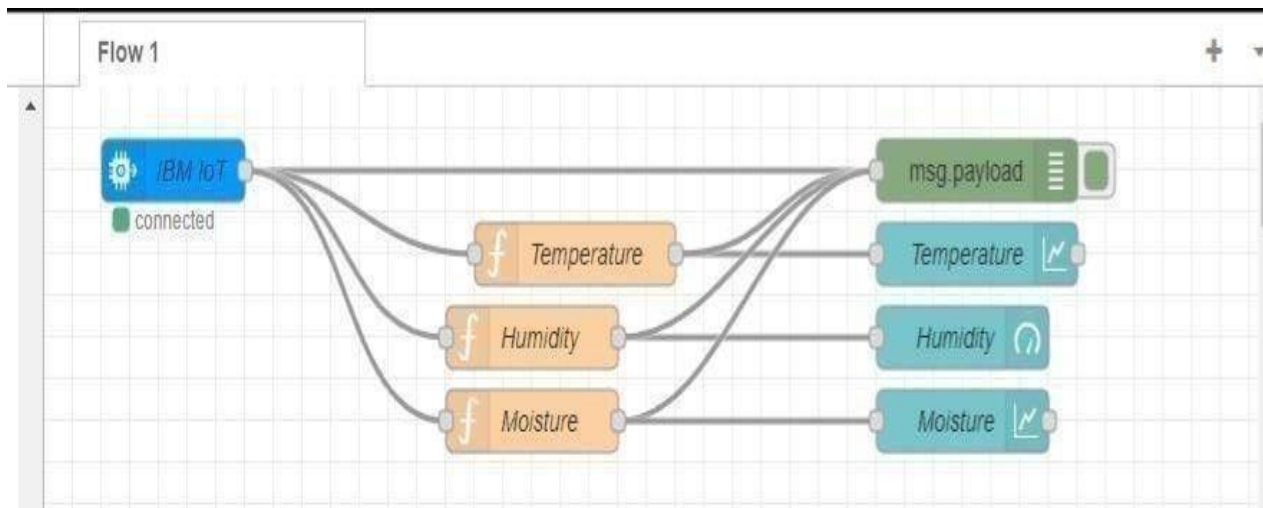
```
msg.payload = msg.payload.d.temperature return
```

```
msg;
```

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



Data received from the cloud in Node-Red console



Nodes connected in following manner to get each reading separately



This is the Java script code I written for the function node to get Temperature separately.

Configuration of Node-Red to collect data from Open Weather

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

HTTP request node is configured with URL we saved before in section 4.4 The data we receive from Open Weather after request is in below JSON

```
format:{"coord":{"lon":79.85,"lat":14.13},"weather":[{"id":803,"main":"Clouds","description":"brokenclouds","icon":"04n"}],"base":"stations","main":{"temp":307.59,"feels_like":305.5,"temp_min":307.59,"temp_max":307.59,"pressure":1002,"humidity":35,"sea_level":1002,"grnd_level":1000},"wind":{"speed":6.23,"deg":170},"clouds":{"all":68},"dt":1589991979,"sys":{"country":"IN","sunrise":1589933553,"sunset":1589979720},"timezone":19800,"id":1270791,"name":"Gūdūr","cod":200}
```

In order to parse the JSON string we use Java script functions and get each parameters

```
var temperature = msg.payload.main.temp;

temperature = temperature-273.15; return

{payload : temperature.toFixed(2)};
```

In the above Java script code we take temperature parameter into a new variable and convert it from kelvin to Celsius

Then we add Gauge and text nodes to represent data visually in UI

