SPRINT -2

TEAM ID :	PNT2022TMID44437
PROJECT NAME:	Smart Farmer-IOT Enabled Smart Farming Application

Connecting IoT Simulator to IBM Watson IoT Platform

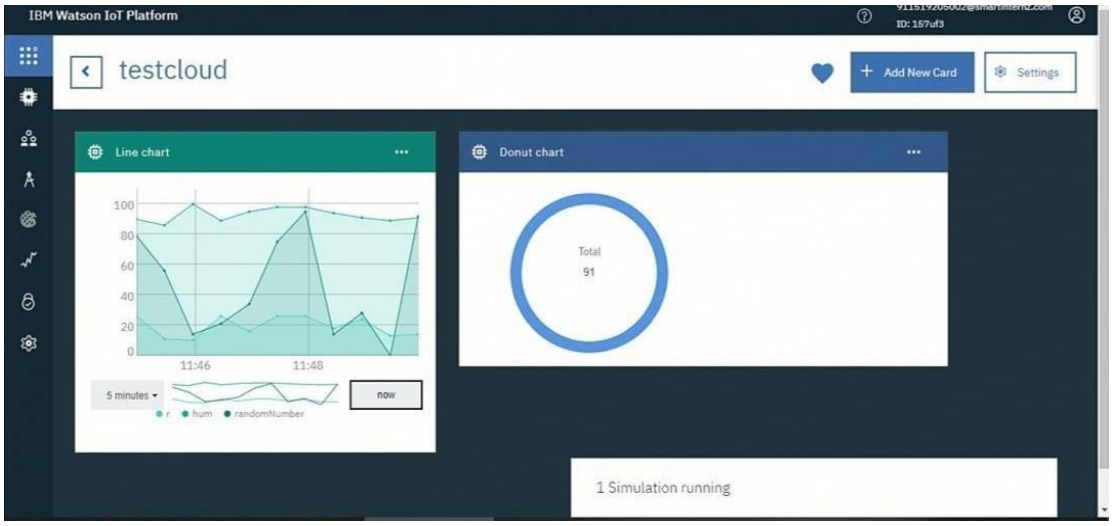
Open link provided in above section 4.3

Give the credentials of your device in IBM Watson IoT

PlatformClick on connect

My credentials given to simulator are:

OrgID: se58mg api: **a-157uf3-f5rg4qxpd3**
Device type:iot**abcde**
Device token:VFBMych&&D)Nu5zeWi
Device ID : 12345



You will see 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 and Data is received in this format **(json).**

{

"e": {

="name": "abce"

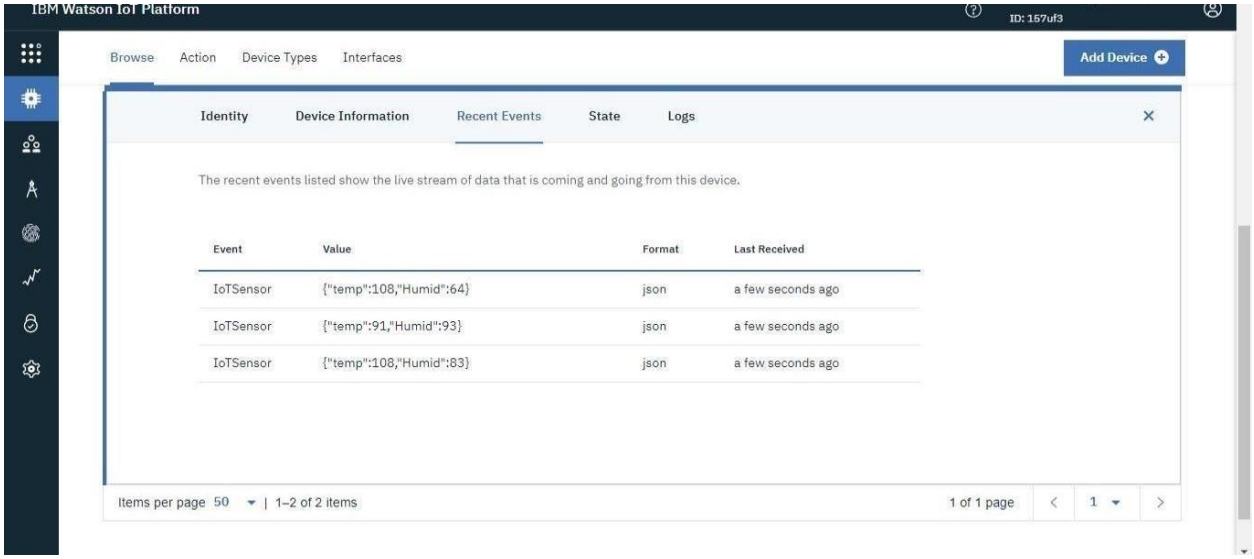
= "temperature": 22,

= "humidity": 69,

="Moisture ": 32

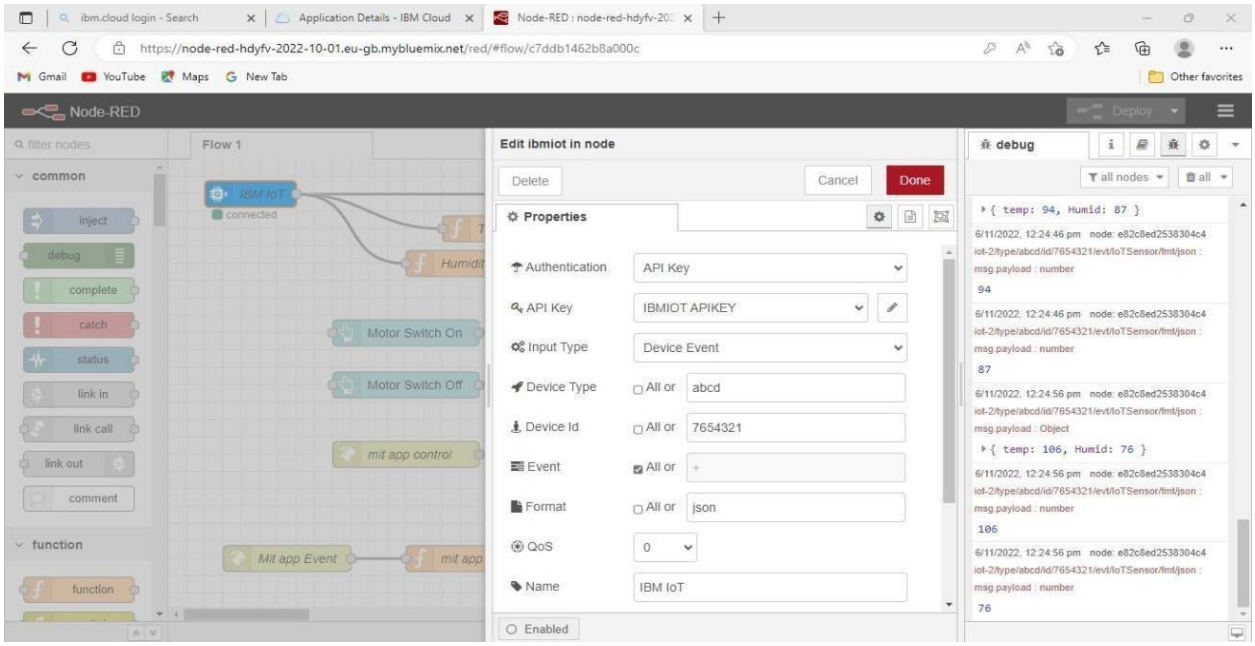
}

}



Configuration of Node-Red to collect IBM cloud data:

- Find the node red starter kit in the IBM cloud catalog.Create your application.
- Enable the continuous delivery feature.
- Open the Node-RED application.Configure your Node-RED application.
- Add extra nodes to your Node-RED palatte.



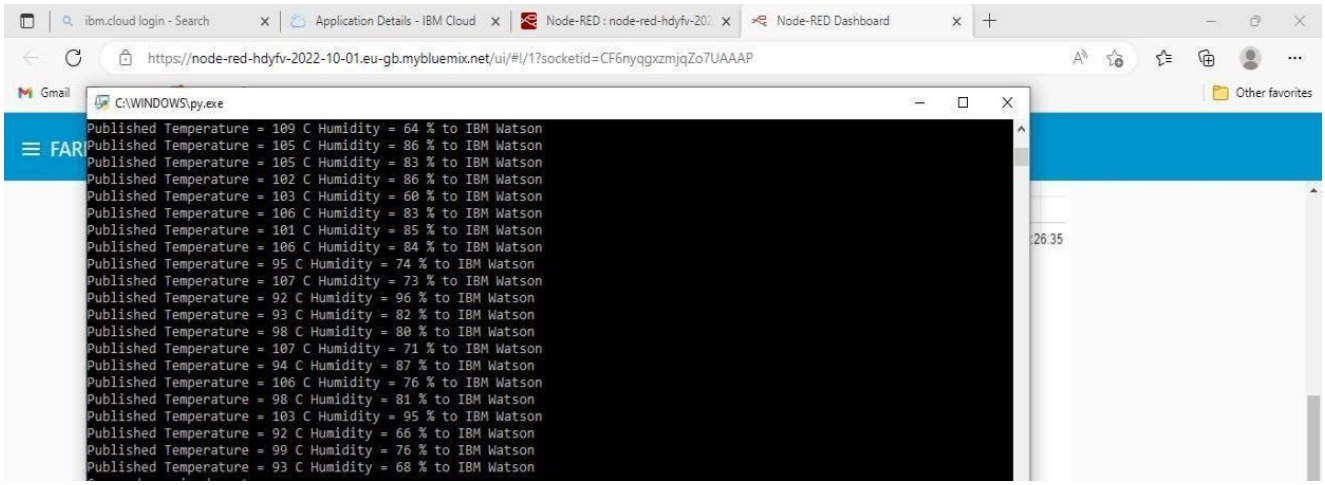
=>Once it is connected Node-Red receives data from the device

=>Display the data using debug node for verification

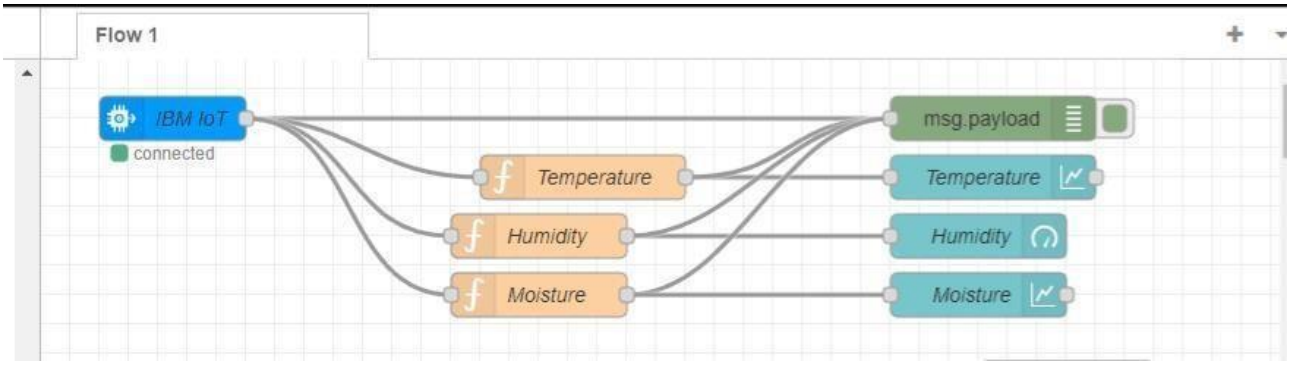
=>Connect function node and write the Java script code to get each reading separately.

=>The Java script code for the function node is:
msg.payload=msg.payload.d.temperature returnmsg;

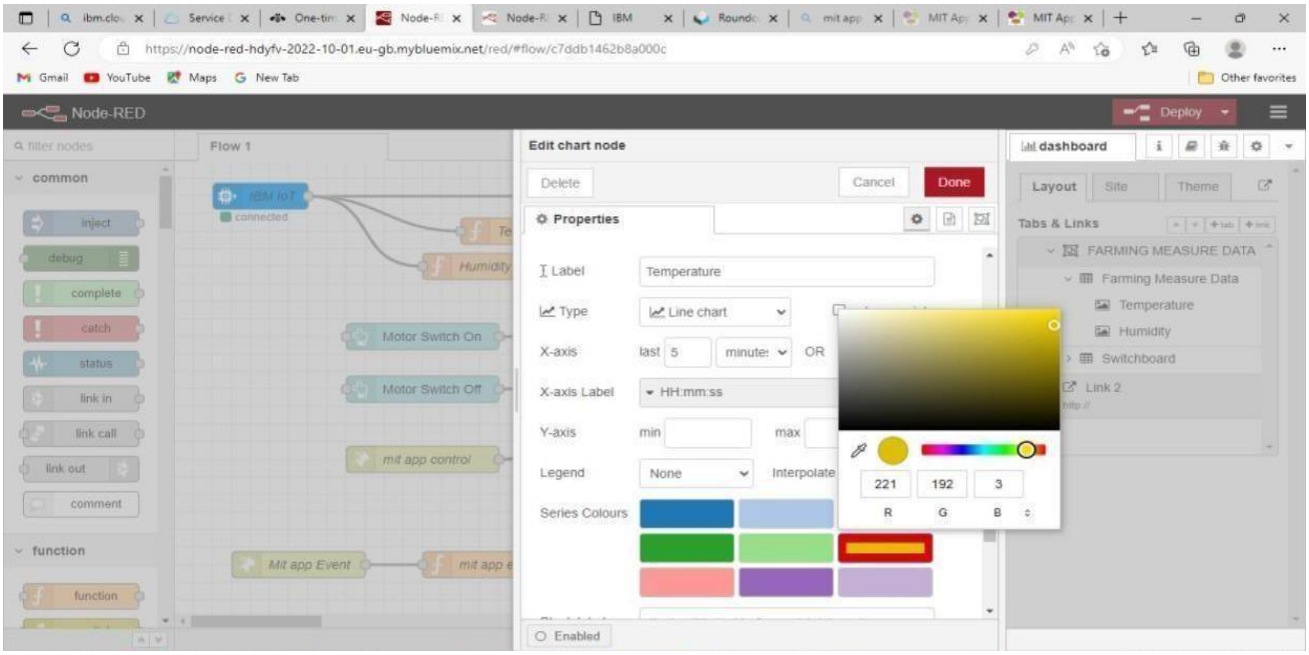
=>Finally connect Gauge nodes from dashboard to see the data in UI.



Data received from cloud in Node-Red console service:



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 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.
HTTP request node is configured with URL we saved before in section 4.4
The

data we receive from OpenWeather 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":30759,"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 variableand
convert it from kelvin to Celsius
Then we add Gauge and text nodes to represent data visually in UI