

IBM NALAIYATHIRAN
SMART FARMER-IOT ENABLED SMART FARMING
APPLICATION

SPRINT 2

Title	Smart farmer-IoT enabled smart farming application
Domain	Internet of Things
Team ID	PNT2022TMID44170
Project Name	Project – Smart Farmer-IoT Enabled smartFarming Application

Building Project

Connecting IoT Simulator to IBM Watson IoT Platform

Open link provided in previous section

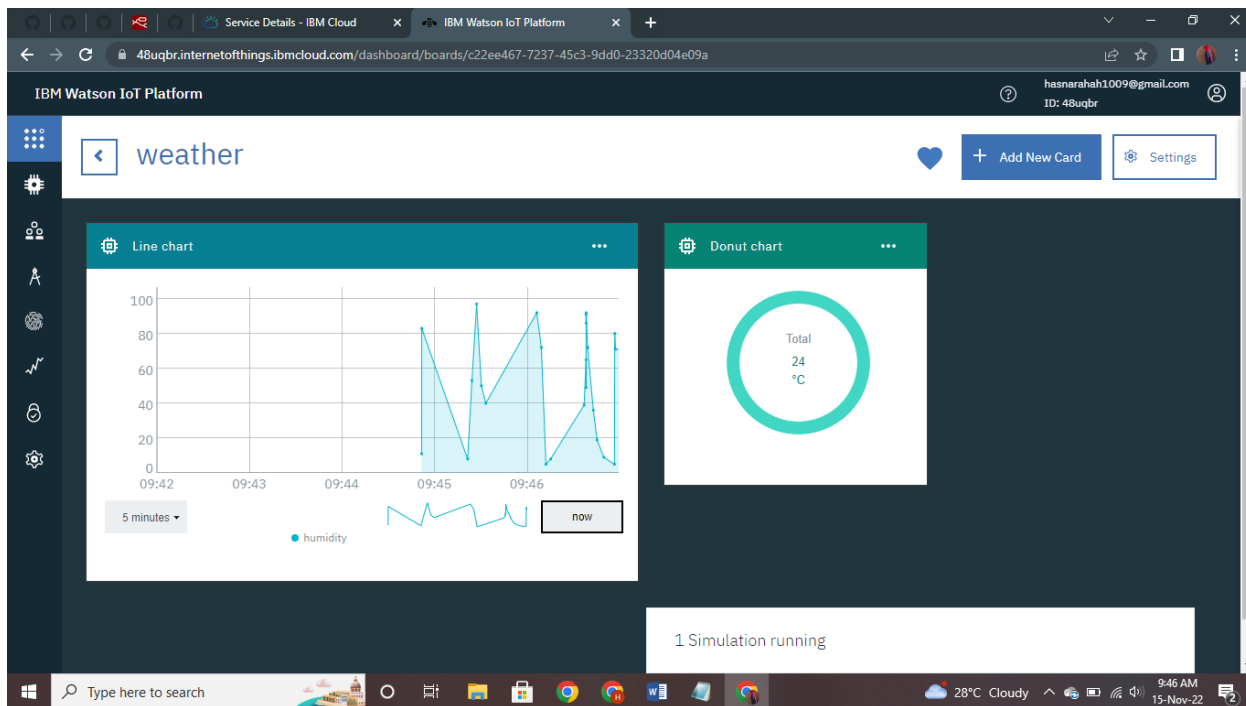
Give the credentials of your device in IBM Watson IoT PlatformClick on
connect

My credentials given to simulator are:

api: a-48uqbr-6lottpkin2

Device type:hasnarahah09

Token: 1XB95kGgzb3cnwT@0u



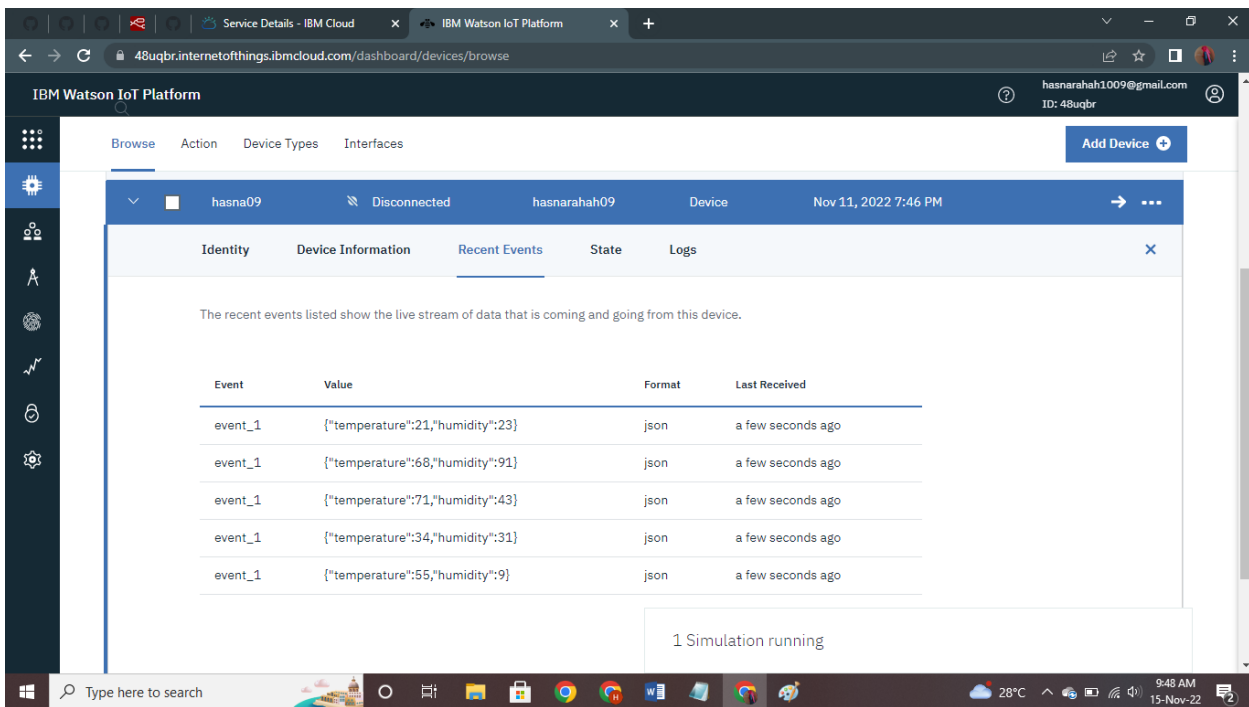
You can see the received data in graphs by creating cards in Boards tab

You will receive the simulator data in cloud

You can see the received data in Recent Events under your device

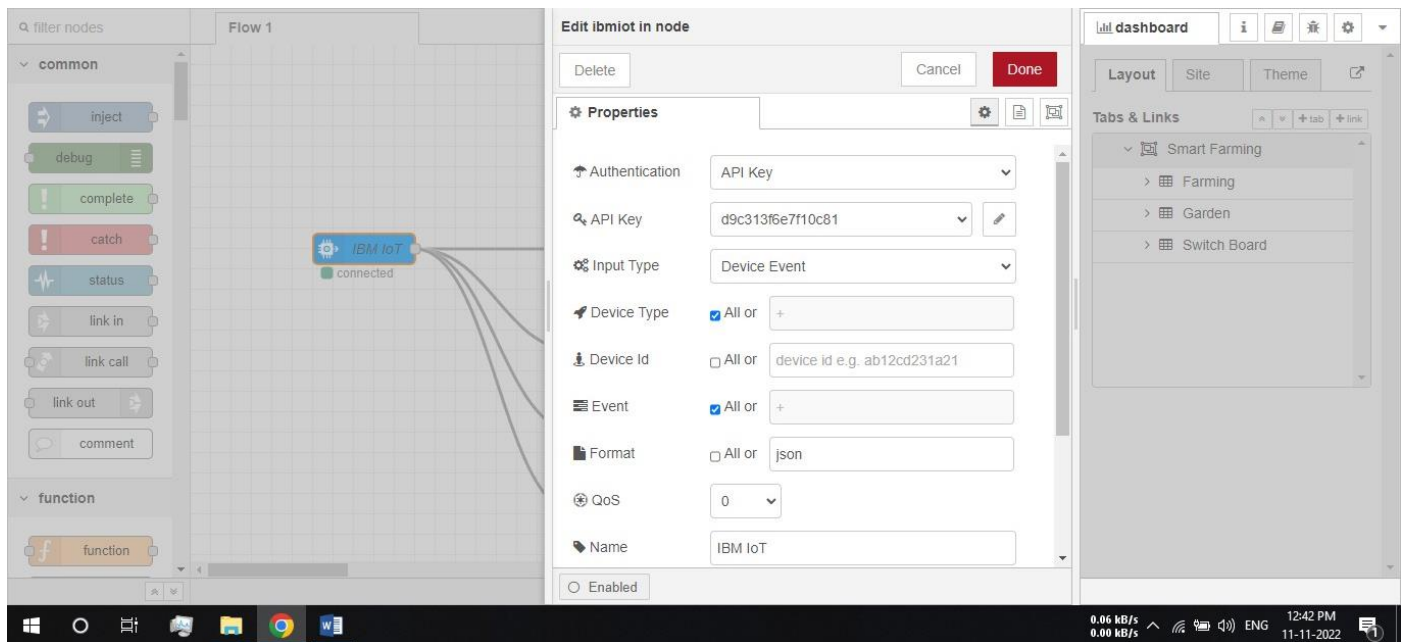
Data received in this format(json)

```
{
  "d": {
    "name": "hasna09",
    "temperature": 17,
    "humidity": 76,
    "Moisture ": 25
  }
}
```



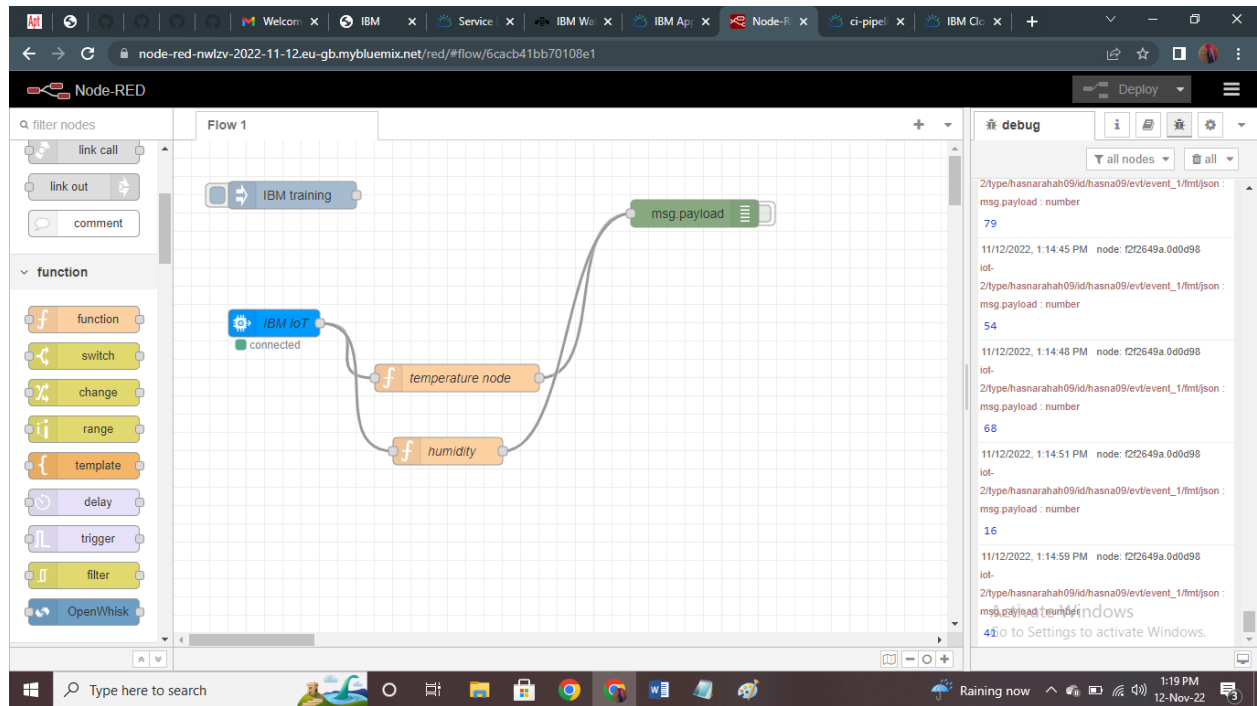
Configuration of Node-Red to collect IBM cloud data

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.

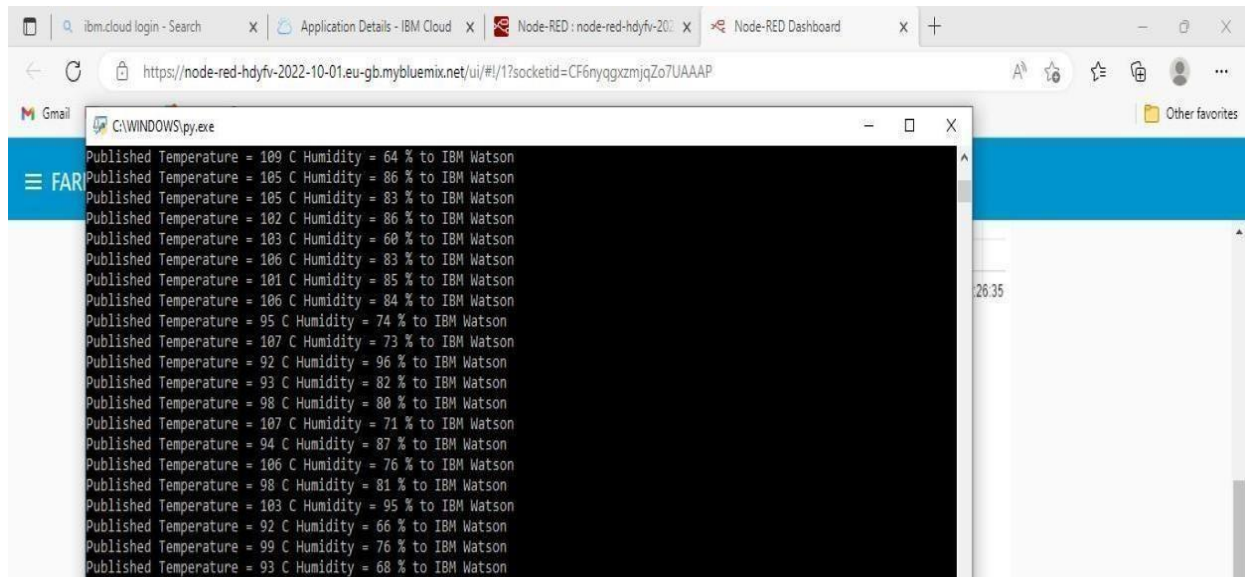


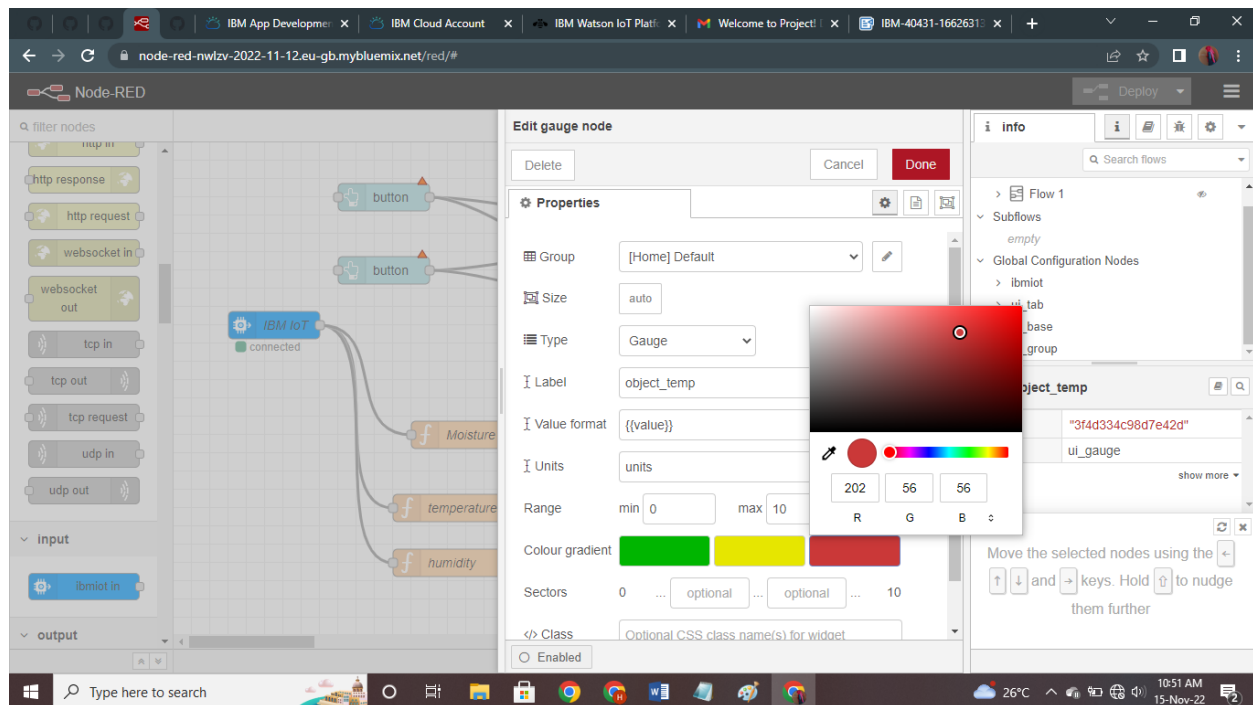
Once it is connected Node-Red receives data from the device Display the data using debug node for verification Connect function node and write the Java script code to get each reading separately.

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





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 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":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

