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	PNT2022TMID05968
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

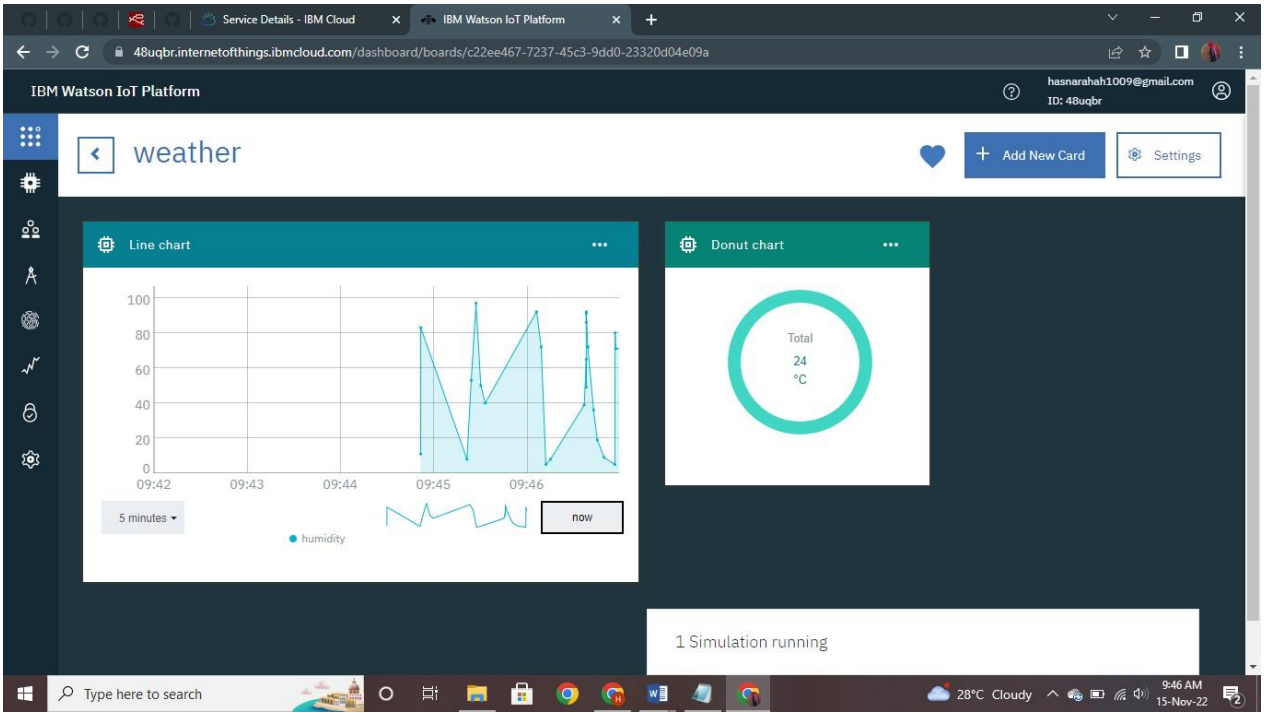
connect

My credentials given to simulator are:

api: a-48uqbr-6lottpkin2

Device type:hasnarahah09

Token: 1XB95kGgzb3cnwT@0u



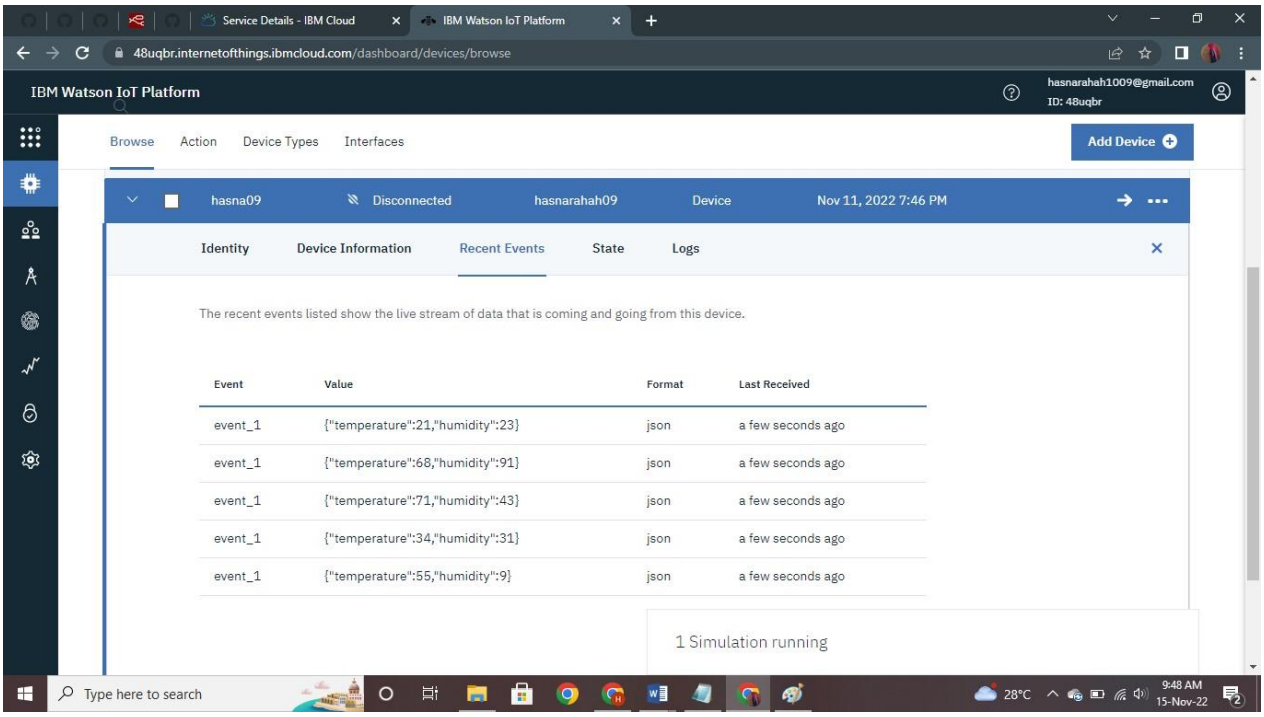
You can see the received data in graphs by creating cards in Boards tab

You will receive the simulator data in cloud

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

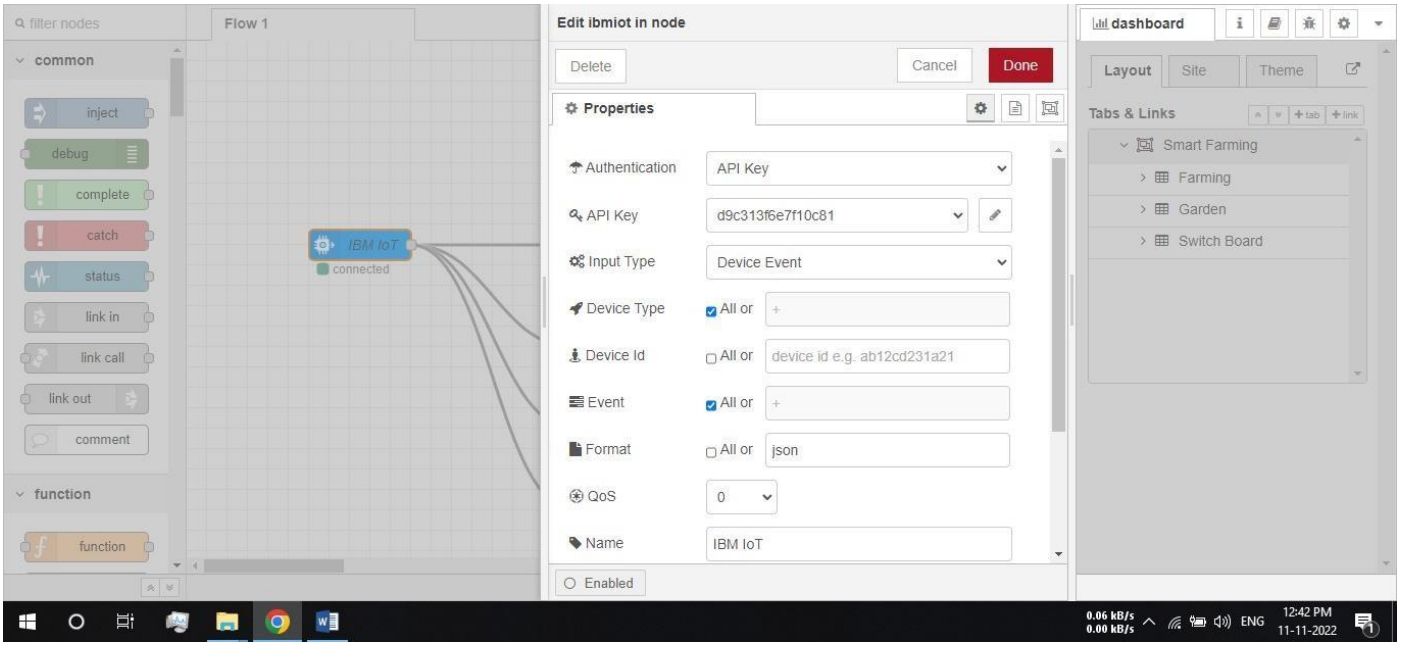
Data received in this format(json)

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

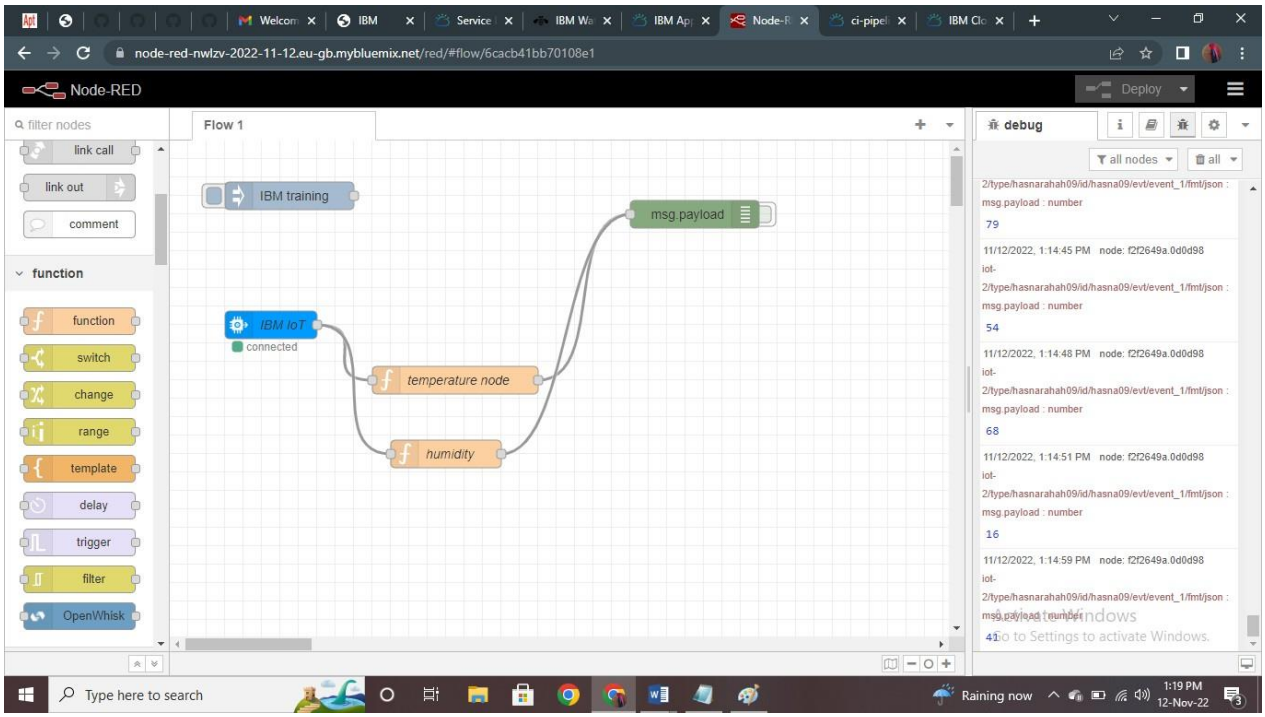


Configuration of Node-Red to collect IBM cloud data

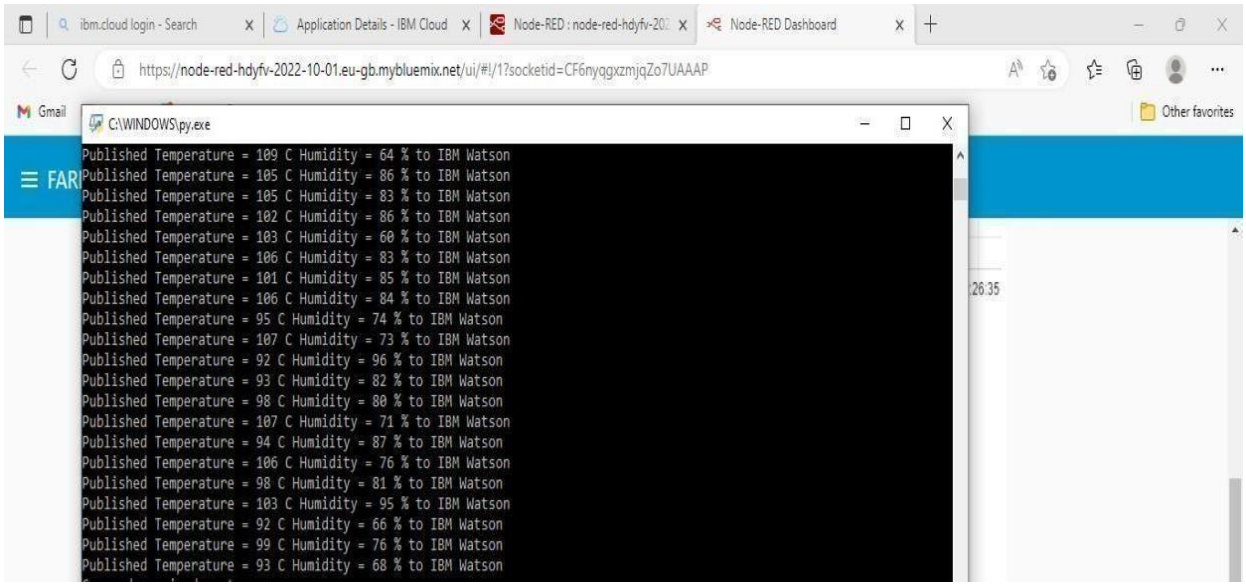
The node IBM IoT App In is added to Node-Red workflow. Then the appropriate device credentials obtained earlier are entered into the node to connect and fetch device telemetry to Node-Red.

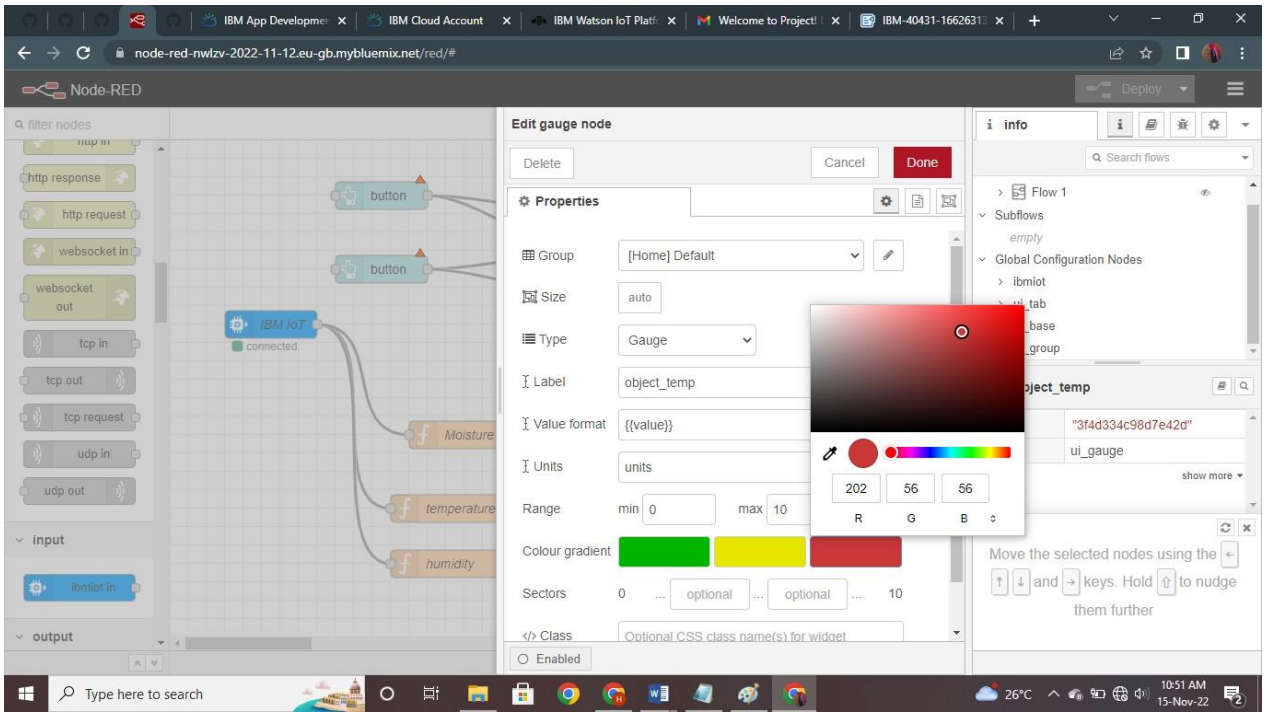


Once it is connected Node-Red receives data from the device Display the data using debug node for verification
Connect function node and write the Java script code to get each reading separately.
The Java script code for the function node is: msg.payload=msg.payload.d.temperature return msg;
Finally connect Gauge nodes from dashboard to see the data in UI



Data received from the cloud in Node-Red console



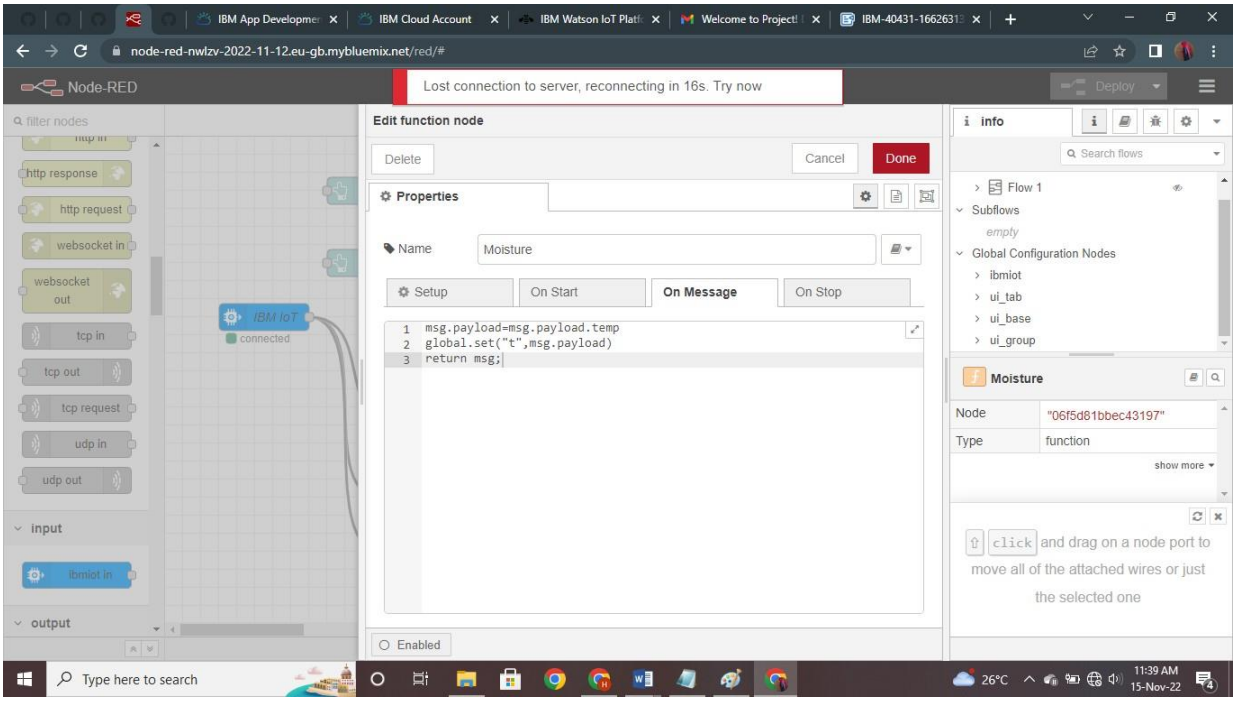


Configuration of Node-Red to collect data from OpenWeather:

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,"h
umidity":35,"sea_level":1002,"grnd_level":1000},"wind":{"speed":6.23,"deg":170
}
,"clouds":{"all":68},"dt":1589991979,"sys":{"country":"IN","sunrise":158993355 3,

"sunset":1589979720},"timezone":19800,"id":1270791,"name":"Gūdūr","cod":20 0}
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


IBM App Developmer
IBM Cloud Account
IBM Watson IoT Platfc
Welcome to Project!
IBM-40431-16626313
node-red-nwlzv-2022-11-12.eu-gb.mybluemix.net/red/#
Node-RED
Lost connection to server, reconnecting in 16s. Try now
Deploy
filter nodes
http response
http request
websocket in
websocket out
tcp in
tcp out
tcp request
udp in
udp out
input
ibmiot in
output
IBM IoT
connected
Edit function node
Delete
Cancel
Done
Properties
Name
Moisture
Setup
On Start
On Message
On Stop
1 msg.payload=msg.payload.temp
2 global.set("t",msg.payload)
3 return msg;
Enabled
info
Search flows
Flow 1
Subflows
empty
Global Configuration Nodes
ibmiot
ui_tab
ui_base
ui_group
Moisture
Node
"06f5d81bbec43197"
Type
function
show more
click and drag on a node port to move all of the attached wires or just the selected one
Type here to search
26°C
11:39 AM
15-Nov-22