

Sprint Delivery – 2

Project	IoT Enabled Smart Farming Application
Team ID	PNT2022TMID26661
Date	09 November 2022

Connecting IOT Simulator to IBM Watson IOTPlatform:

Give the credentials of your device in IBM Watson IOTPlatform Click on connect My credentials given to simulator are:

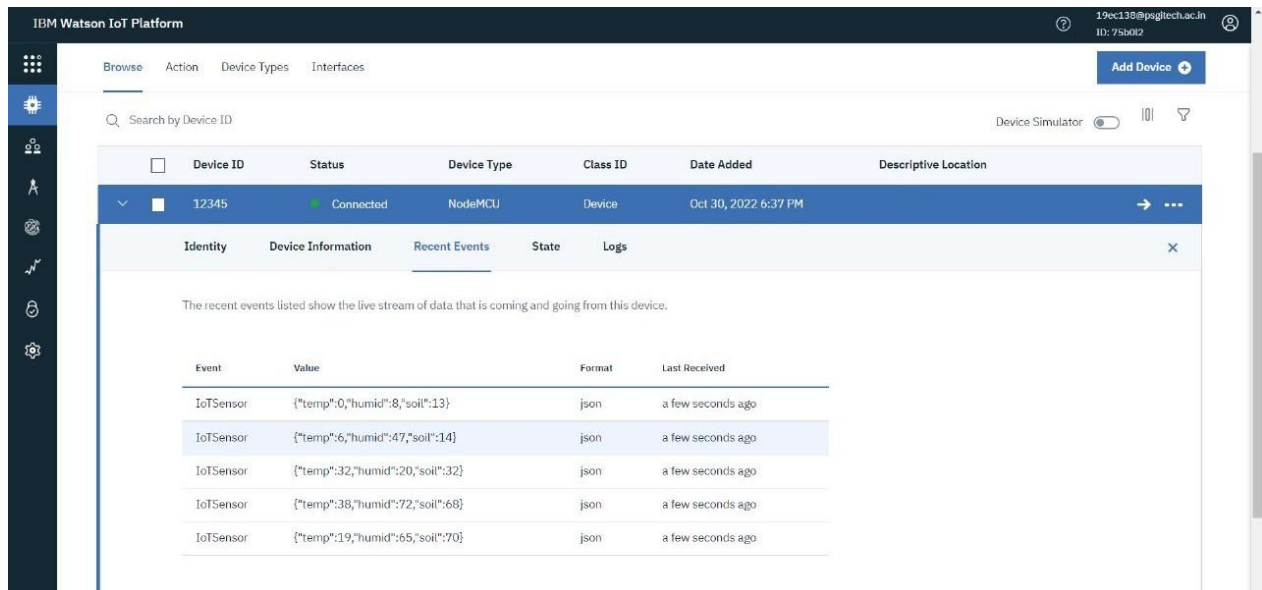
OrgID: 75b012

Device type : NodeMCU

Authentication method : use-token-auth

Device ID : 12345

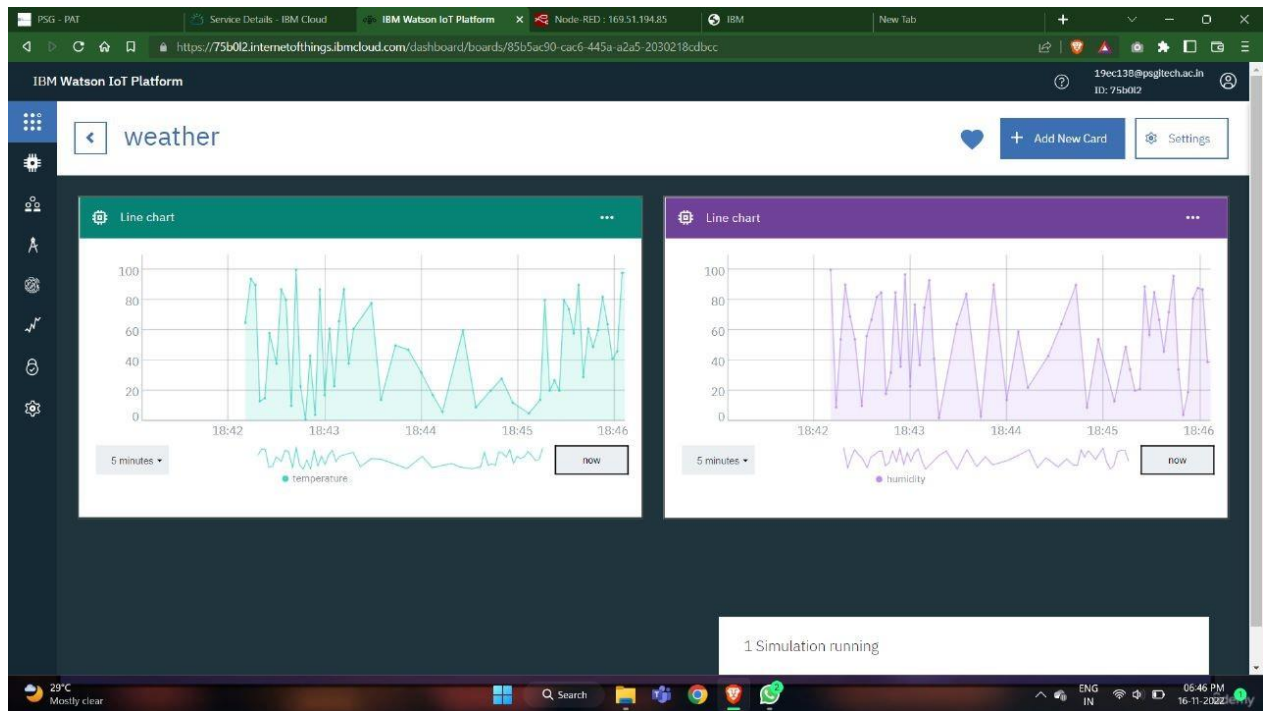
Authentication Token : 12345678



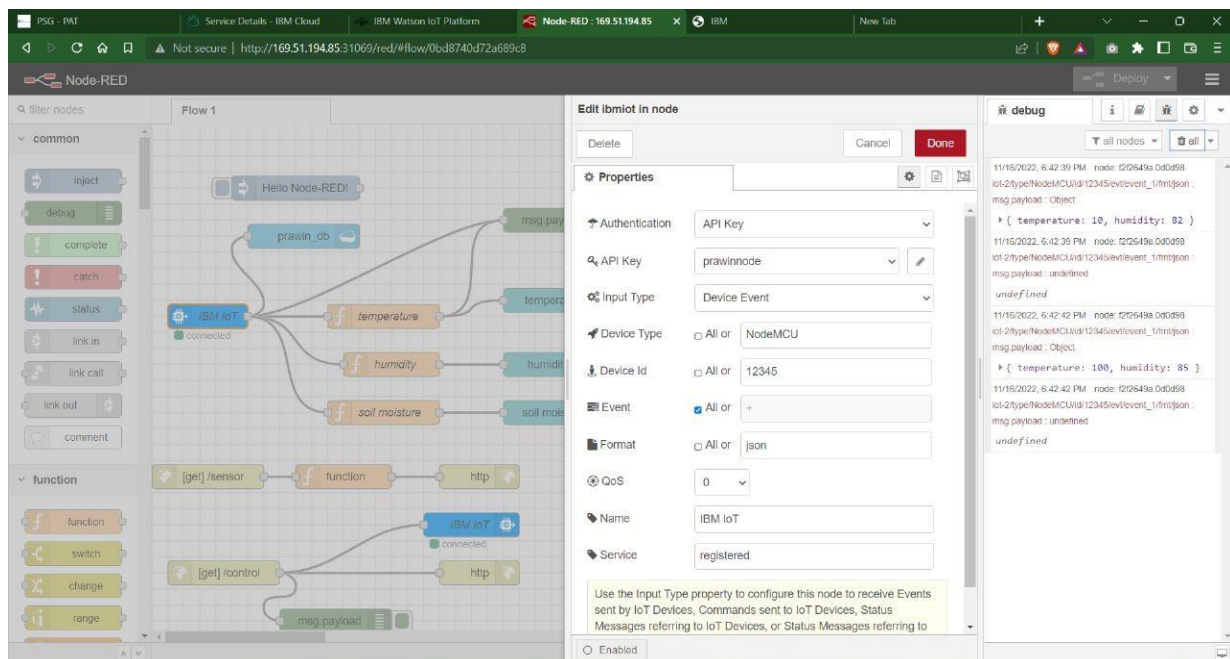
The screenshot displays the IBM Watson IoT Platform interface. At the top, the header shows the user's email (19ed138@psgtechnic.in) and the organization ID (75b012). The main navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar is present with the text 'Search by Device ID'. A table lists devices, with the first device '12345' highlighted. This device is of type 'NodeMCU' and is 'Connected'. Below the table, the 'Recent Events' tab is selected, showing a live stream of data. The data is presented in a table with columns: Event, Value, Format, and Last Received. The events are JSON payloads from an 'IoTSensor'.

Event	Value	Format	Last Received
IoTSensor	{"temp":0,"humid":8,"soil":13}	json	a few seconds ago
IoTSensor	{"temp":6,"humid":47,"soil":14}	json	a few seconds ago
IoTSensor	{"temp":32,"humid":20,"soil":32}	json	a few seconds ago
IoTSensor	{"temp":38,"humid":72,"soil":68}	json	a few seconds ago
IoTSensor	{"temp":19,"humid":65,"soil":70}	json	a few seconds ago

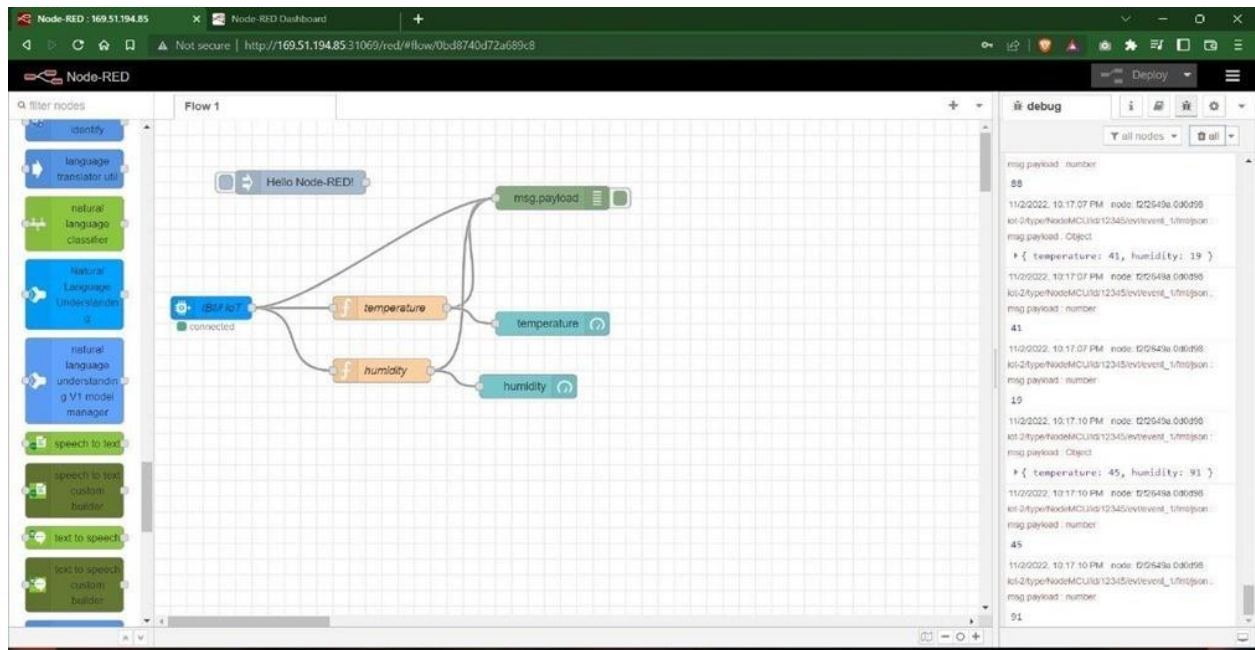
For initial testing , a node MCU device is created in IBM Watson IOT platform .Further the simulation is made run and Json data is seen in recent events of the same device. Also the graph has been visualized.



For the next process, the randomly generated values should be displayed in Node Red. So the Node red is configured.



Configuring Node-RED:



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

