

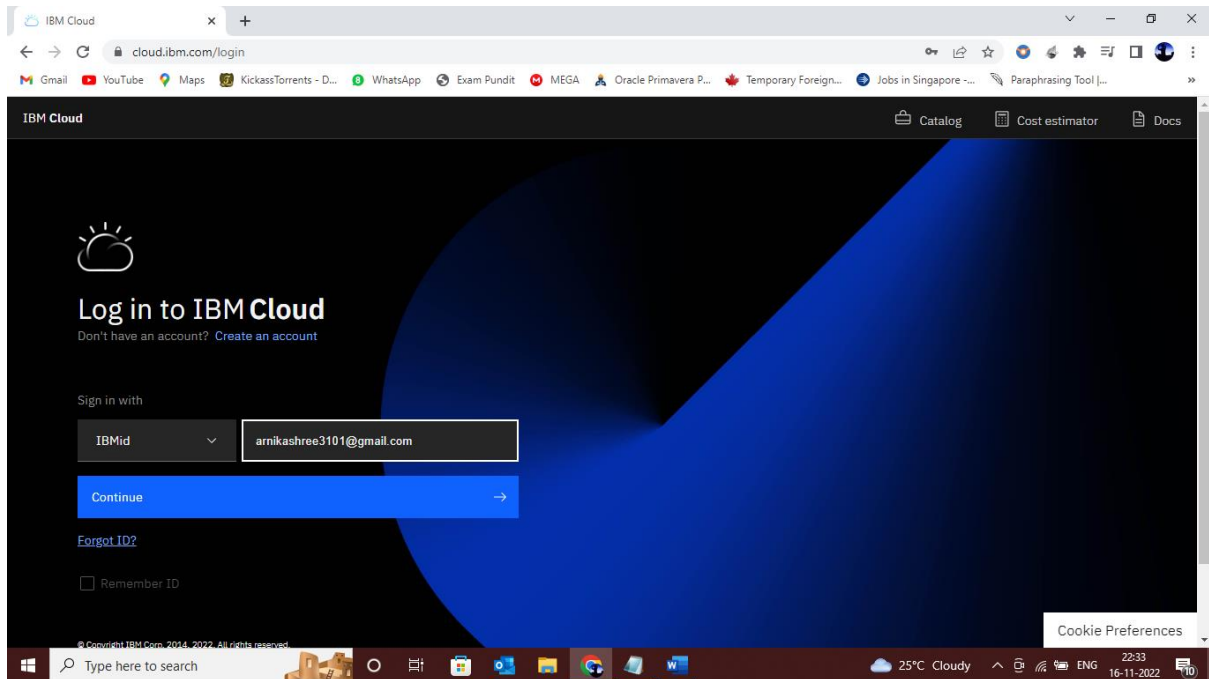
# Project Development Phase

## Delivery of Sprint -2

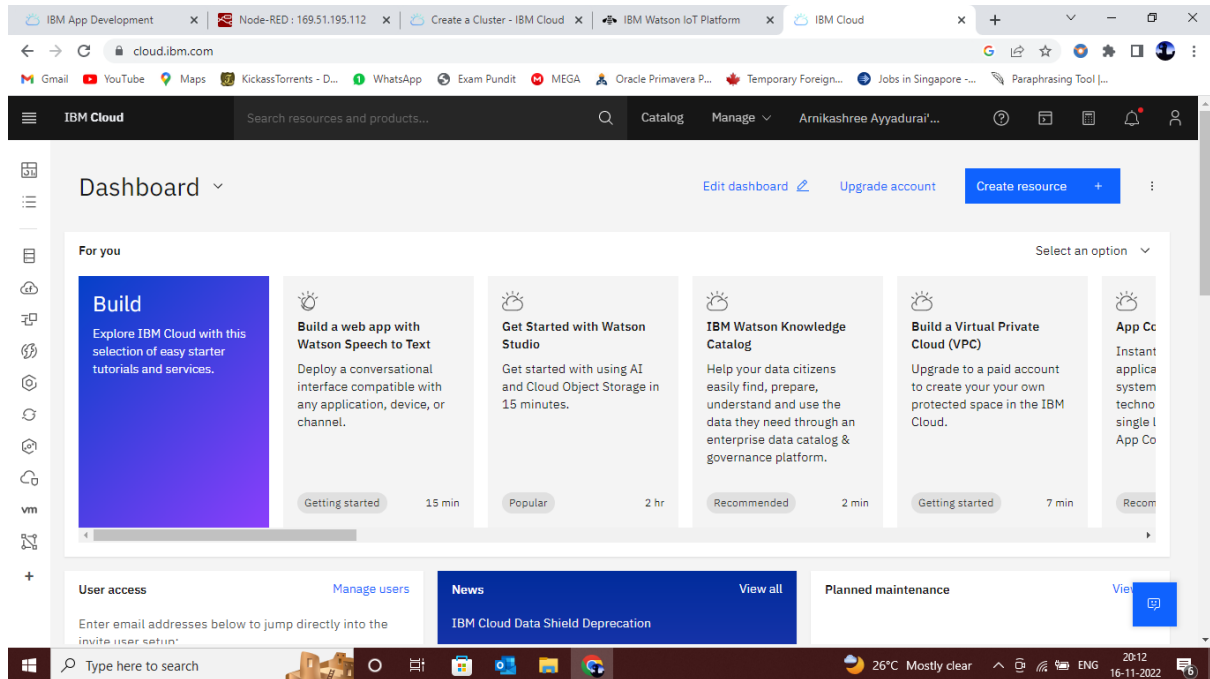
<b>Team ID</b>	PNT2022TMID48383
<b>Project Name</b>	Smart Farmer-IOT Enabled Smart Farming Application

In Sprint-2 we are going to develop the IBM Watson and making the connection to the node-red.

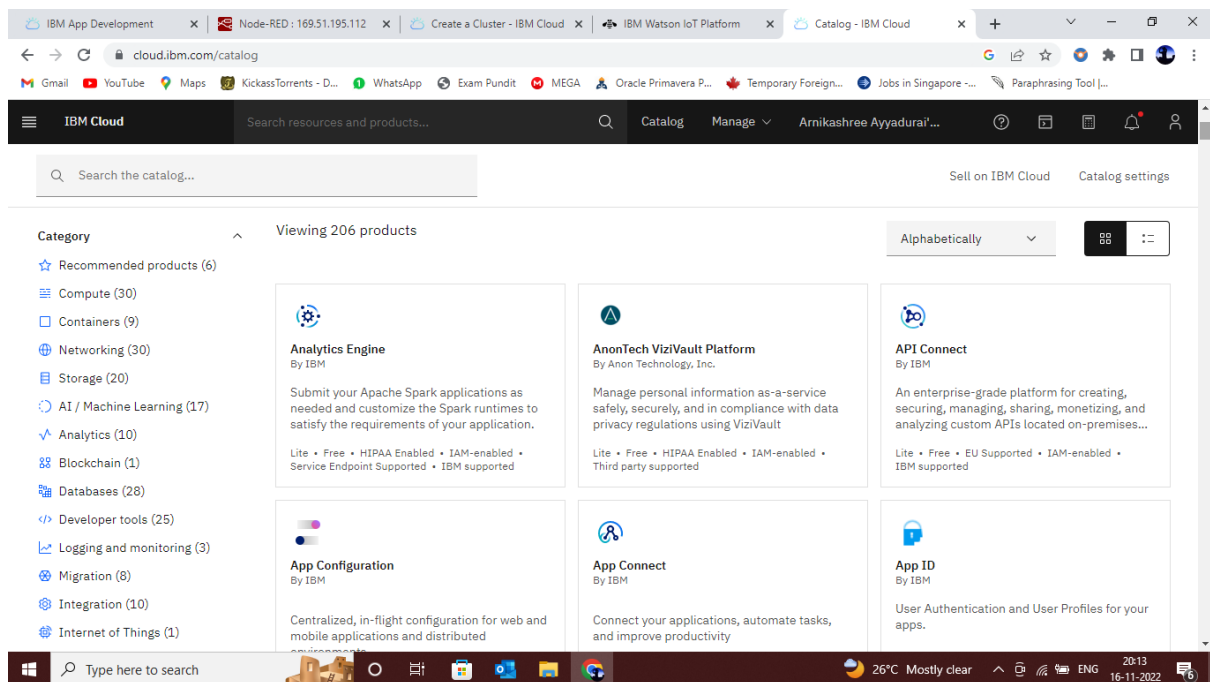
### Login into IBM cloud:



Click the Catalog button:



After clicking the catalog .Select the Internet of Things and then click.



If you have already existing plan we can continue or we have to create new one.

The screenshot shows the IBM Cloud service catalog for the Internet of Things Platform. The page is titled "Internet of Things Platform" and includes a description: "This service is the hub of all things IBM IoT, it is where you can set up and manage your connected devices so that your apps can access their live and historical data." The "Create" tab is active, showing a "Select a location" dropdown set to "London (eu-gb)" and a "Select a pricing plan" section. A table lists the available plans:

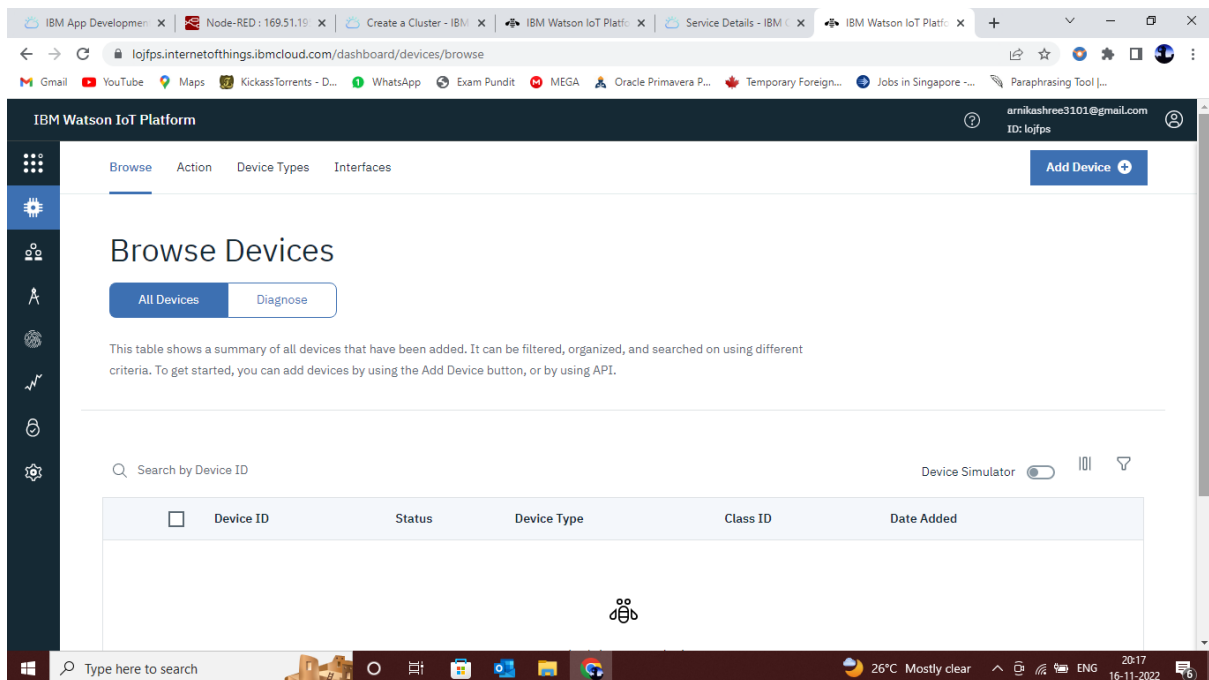
Plan	Features	Pricing
Lite	Includes up to 500 registered devices, and a maximum of 200 MB of each data metric Maximum of 500 registered devices Maximum of 500 application bindings	Free

The right sidebar shows a "Summary" section with details: "Internet of Things Platform", "Free", "Location: London", "Plan: Lite", "Service name: Internet of Things Platform-uf", and "Resource group: Default". There are buttons for "Create" and "Add to estimate".

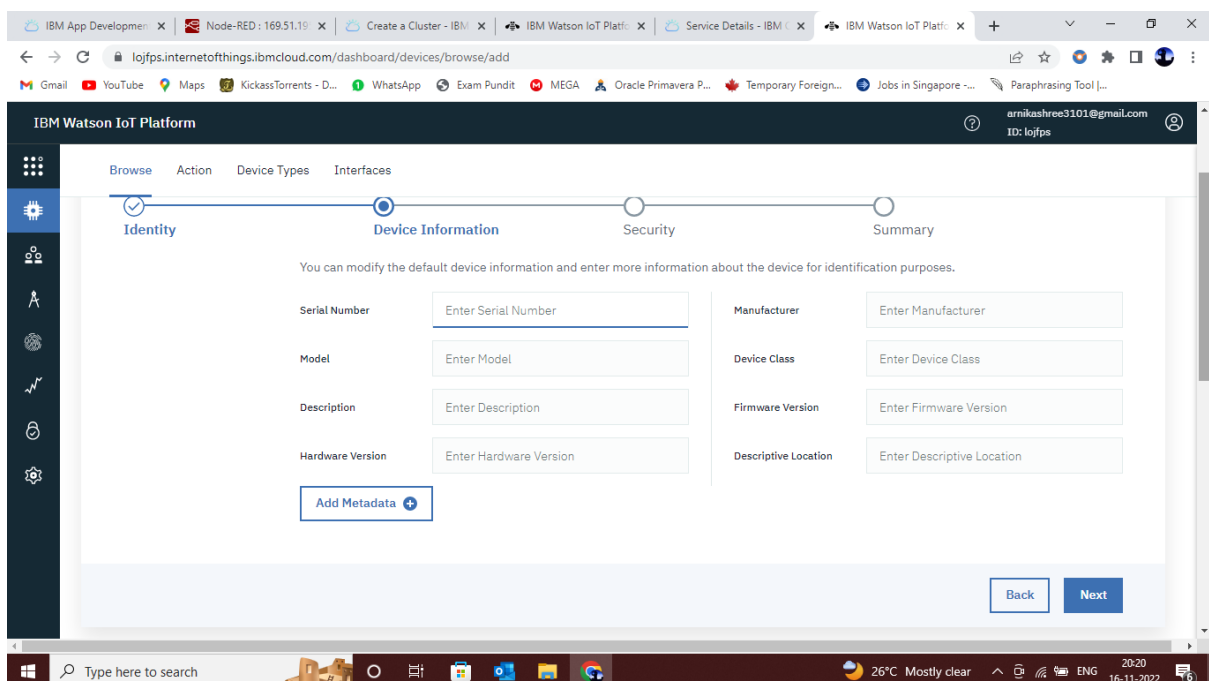
Next window will be appear after clicking the exiting plan and click the launch button.

The screenshot shows the "Service Details" page for the Internet of Things Platform. The page is titled "Internet of Things Platform-i3" and includes a "Launch" button. The "Manage" tab is active, showing a "Plan" section with a "Launch" button. The "Connections" section is also visible. The page includes a "Let's get started with IBM Watson IoT Platform" section with a "Launch" button and a "Docs" button. Below this, there is a "Ready for the next level?" section with a "IBM Watson IoT Platform Journey" section. The journey is shown as a sequence of steps: "Lite" (completed), "Non-Production", and "Production".

Now we have to create the Device types and click the add device type button.



After clicking we can see the window like these and press finish button.



Now we have to Register Device in the Watson platform. click register device button.

The screenshot shows the IBM Watson IoT Platform interface. The 'Browse Devices' page is active, displaying a table of registered devices. The table has columns for Device ID, Status, Device Type, Class ID, and Date Added. One device, 'Weather\_today', is listed with a status of 'Disconnected' and a type of 'Weather'. The 'Add Device' button is located in the top right corner of the interface.

Device ID	Status	Device Type	Class ID	Date Added
Weather_today	Disconnected	Weather	Device	Nov 16, 2022 8:22 PM

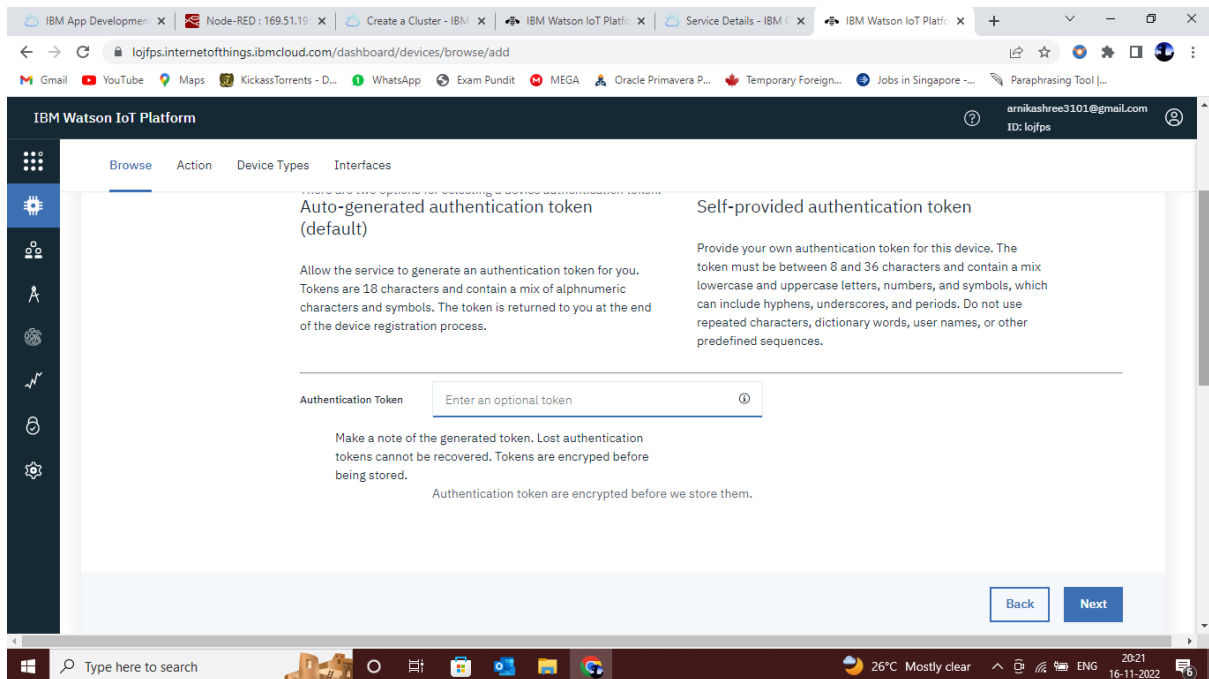
Type the device type and device id and then click the next button.

The screenshot shows the 'Add Device' wizard in the IBM Watson IoT Platform. The 'Identity' step is selected, and the user is prompted to select a device type and provide a unique ID. The 'Device Type' field is set to 'Weather' and the 'Device ID' field is set to 'Weather\_today'. The 'Next' button is visible at the bottom right of the wizard.

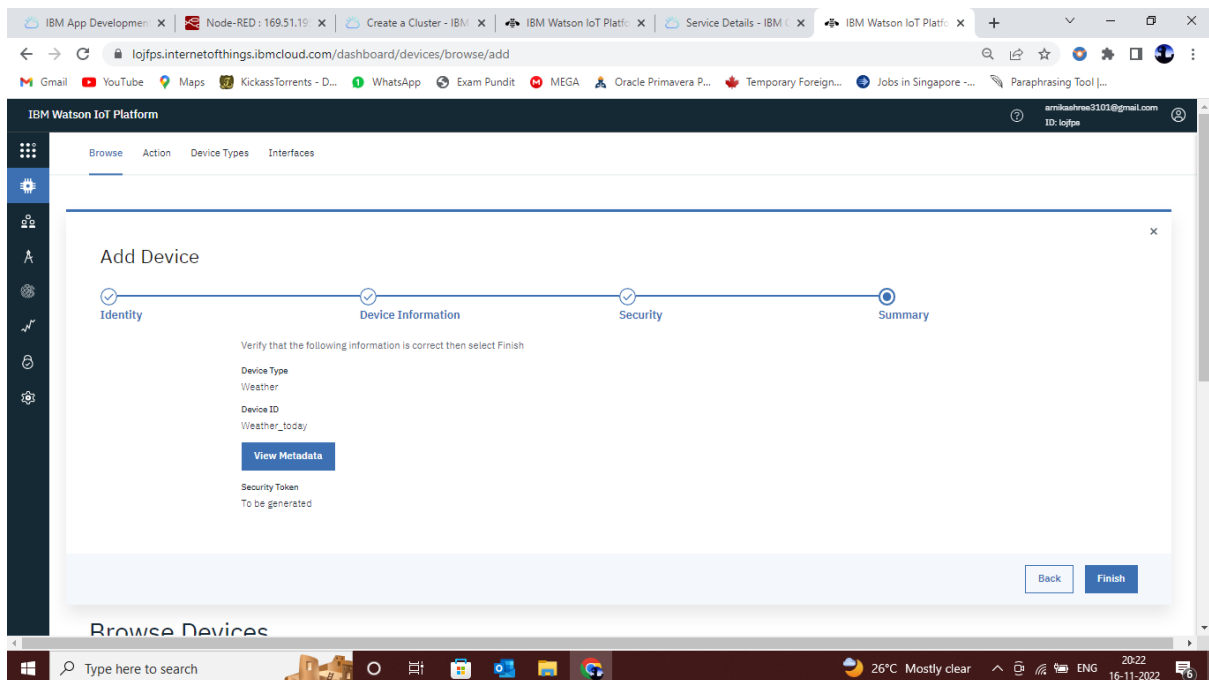
Device Type: Weather

Device ID: Weather\_today

Now we have to give one authentication token that token is more than 8 characters and below 36 characters.



Then click the finish button.



Now it will generate the credentials that credentials are very important further connecting to node-red and in python code or simulation. so we have to save in notepad.

## My Device Credentials:

Organization ID - lojfps

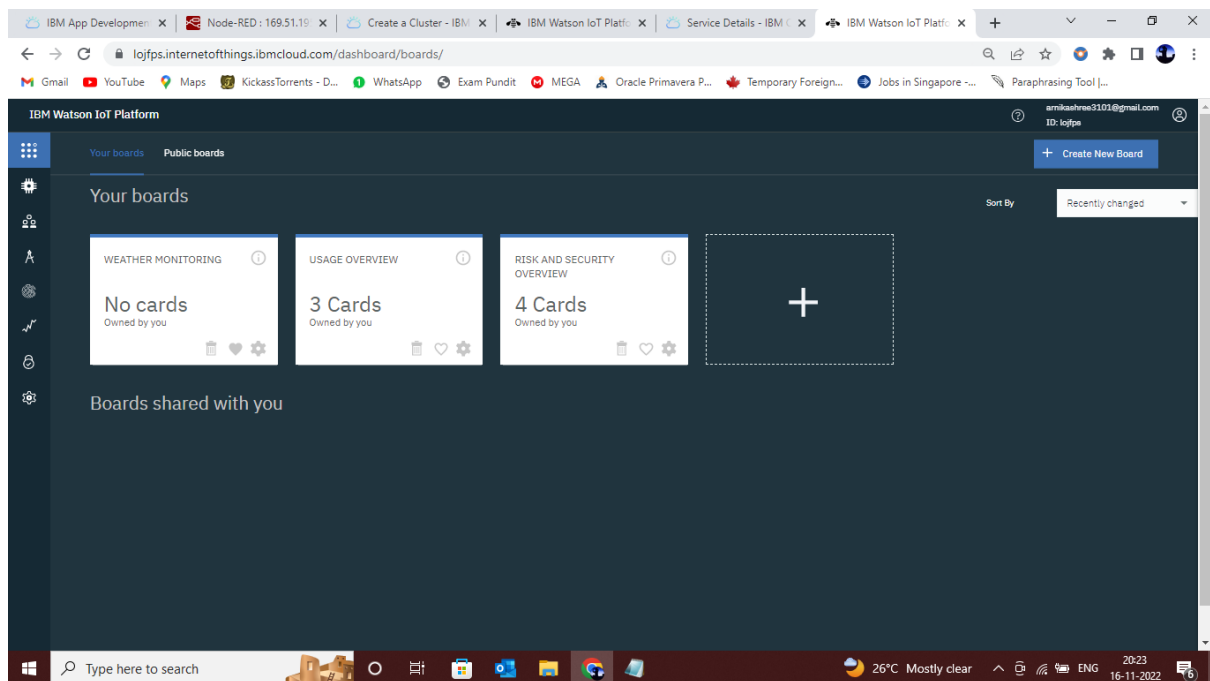
Device Type - Weather

Device ID - Weather\_today

Authentication Method - use-token-auth

Authentication Token - 9LLe)1F)Sjlj157f\*(

Now we have to add boards. Because we can data as graph model.



After adding boards we can run simulation and see the simulation as shown below. You will receive the simulator data in cloud .

The screenshot shows the IBM Watson IoT Platform interface. The main dashboard is titled "Weather Monitoring" and features a donut chart labeled "Donut chart" with a value of "Total 21 °C". A sidebar on the left contains navigation icons. A panel on the right, titled "Simulations", shows "2/50 Simulations Running" and a list of devices: "Weather\_today" and "Weather\_1". Below the list are buttons for "Create Simulated Device" and "Use Registered Device". At the bottom of the simulation panel, it indicates "132 events sent" and "3.15 KB sent". The browser's address bar shows the URL "lojfps.internetofthings.ibmcloud.com/dashboard/boards/9e48b584-e766-40b7-ba39-379a46746d5e". The Windows taskbar at the bottom shows the time as 21:04 on 16-11-2022.

Now getting the random temperature values in the IBM Watson. You can see the received data in Recent Events under your device. So finally we can generate temperature values as like real sensors.

The screenshot shows the IBM Watson IoT Platform interface, specifically the "devices/browse" page. The page displays a table of devices with columns: Device ID, Status, Device Type, Class ID, and Date Added. The device "Weather\_1" is listed with a status of "Connected" and a date added of "Nov 16, 2022 8:57 PM". Below the table, there is a section for "Recent Events" for the selected device. This section contains a table with columns: Event, Value, Format, and Last Received. The table lists five events, each with a value of {"temperature": [random number]} and a format of "json". The last received time for all events is "a few seconds ago". A notification at the bottom right of the page states "2 Simulations running". The browser's address bar shows the URL "lojfps.internetofthings.ibmcloud.com/dashboard/devices/browse". The Windows taskbar at the bottom shows the time as 21:11 on 16-11-2022.

Event	Value	Format	Last Received
event_1	{"temperature":90}	json	a few seconds ago
event_1	{"temperature":9}	json	a few seconds ago
event_1	{"temperature":34}	json	a few seconds ago
event_1	{"temperature":34}	json	a few seconds ago
event_1	{"temperature":79}	json	a few seconds ago

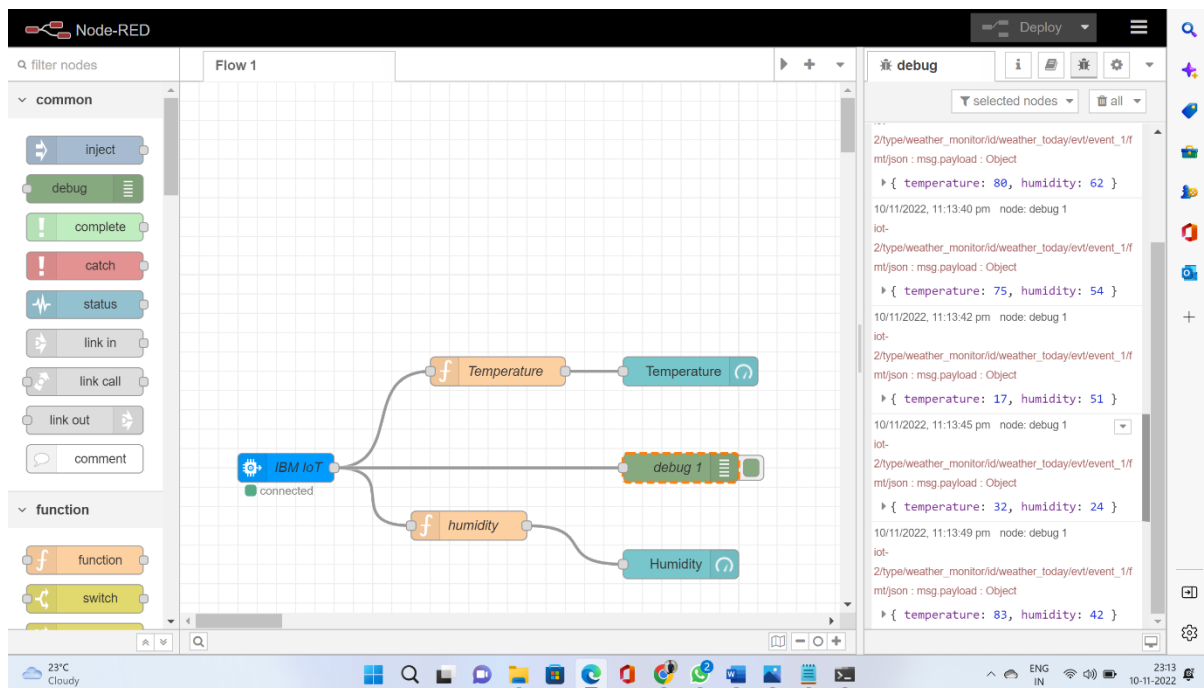


## Now Configuration the Node-Red with IBM Watson Platform to collect the IBM cloud data:

The IBM Watson 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.

**Api key** - a-lojfps-pt8ozsandy

**Authentication Token** - NCmrA9oc\*LPpz-ZnAG



Once it is connected to the Node-Red it receives the data from the Watson. Displaying the data using debug node in the left side of the workspace. And also see the results in the debug node

Node-RED

filter nodes

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

function

switch

Edit function node

Delete

Cancel

Done

Properties

Name

Temperature

Setup

On Start

On Message

On Stop

1

msg.payload= msg.payload.temperature

2

global.set('t',msg.payload)

3

return msg;

Enabled

debug

selected nodes

all

2/type/weather\_monitor/id/weather\_today/evt/event\_1/fmt/json : msg.payload : Object

> { temperature: 32, humidity: 24 }

10/11/2022, 11:13:49 pm node: debug 1

iot-

2/type/weather\_monitor/id/weather\_today/evt/event\_1/fmt/json : msg.payload : Object

> { temperature: 83, humidity: 42 }

10/11/2022, 11:13:52 pm node: debug 1

iot-

2/type/weather\_monitor/id/weather\_today/evt/event\_1/fmt/json : msg.payload : Object

> { temperature: 34, humidity: 95 }

10/11/2022, 11:13:54 pm node: debug 1

iot-

2/type/weather\_monitor/id/weather\_today/evt/event\_1/fmt/json : msg.payload : Object

> { temperature: 58, humidity: 17 }

10/11/2022, 11:13:57 pm node: debug 1

iot-

2/type/weather\_monitor/id/weather\_today/evt/event\_1/fmt/json : msg.payload : Object

> { temperature: 15, humidity: 10 }

Connect function node and write the Java script code to get each reading separately.  
Function node is rename as the temperature and humidity. And write json code on message.

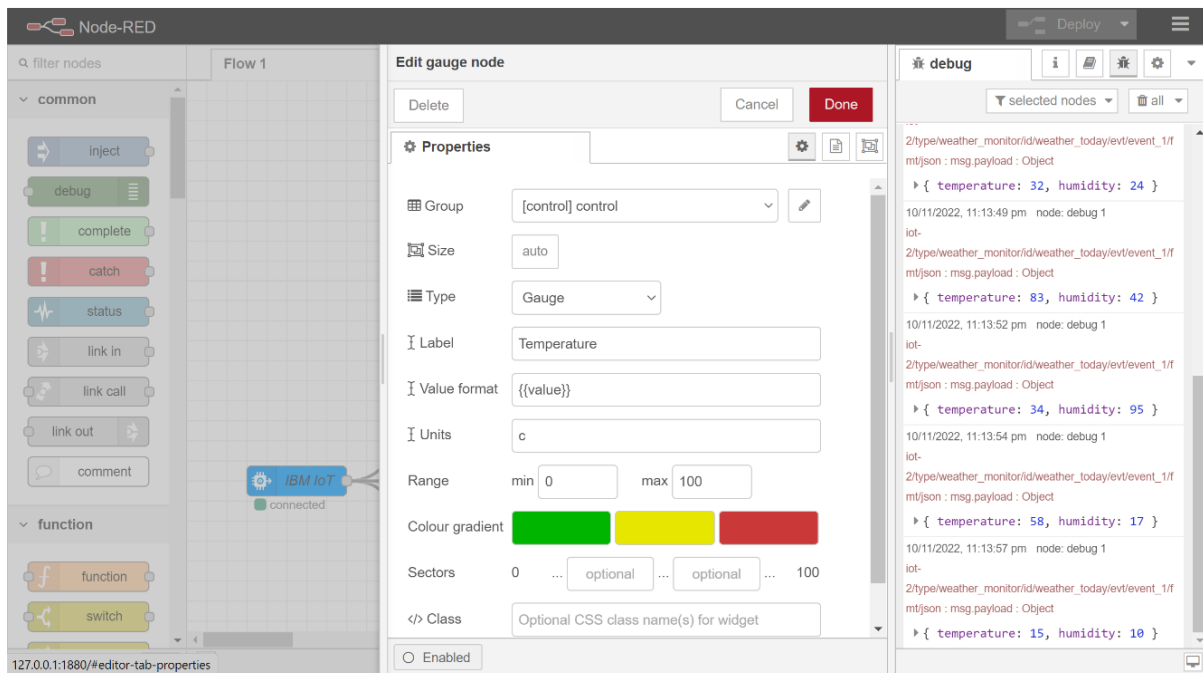
### Json code for Temperature:

```
msg.payload= msg.payload.temperature  
global.set('t',msg.payload)  
return msg.
```

### Json code for Humidity:

```
msg.payload= msg.payload.humidity  
global.set('h',msg.payload)  
return msg.
```

Finally connect the Gauge nodes from node-red to see the data in the node-red dashboard UI:



Now we can see the output in the node-red dashboard.

