

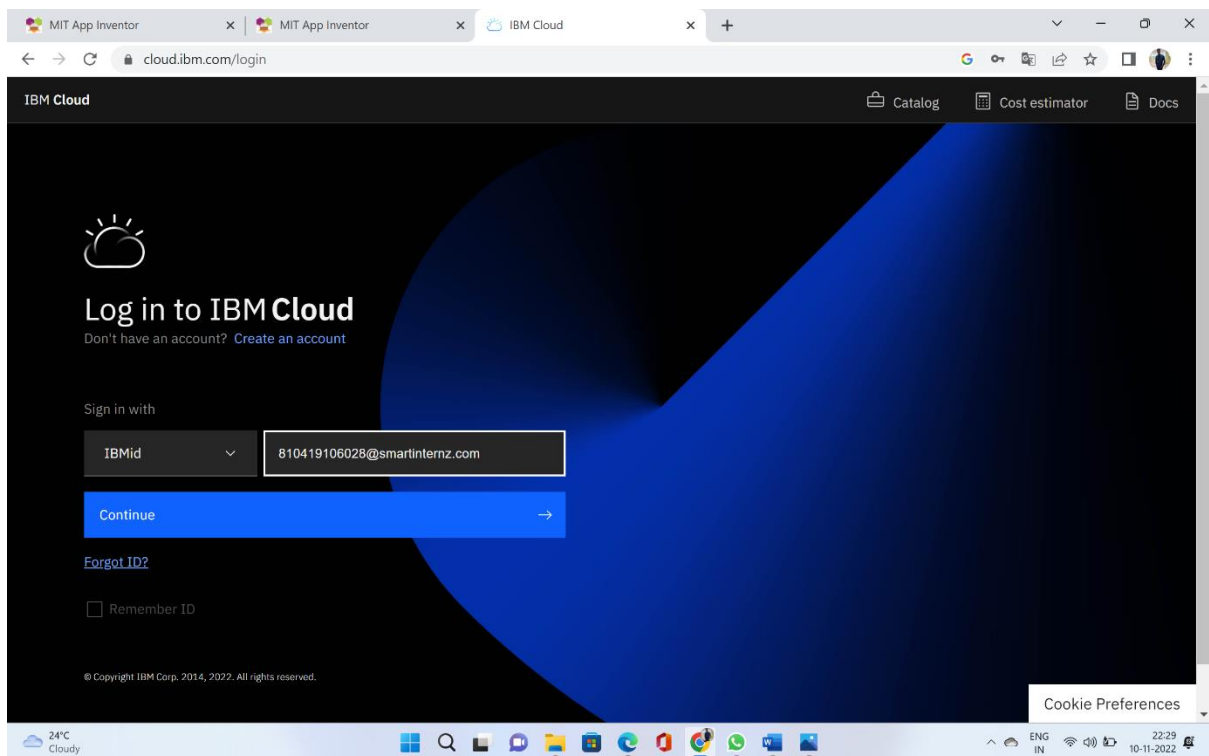
Project Development Phase

Delivery of Sprint -2

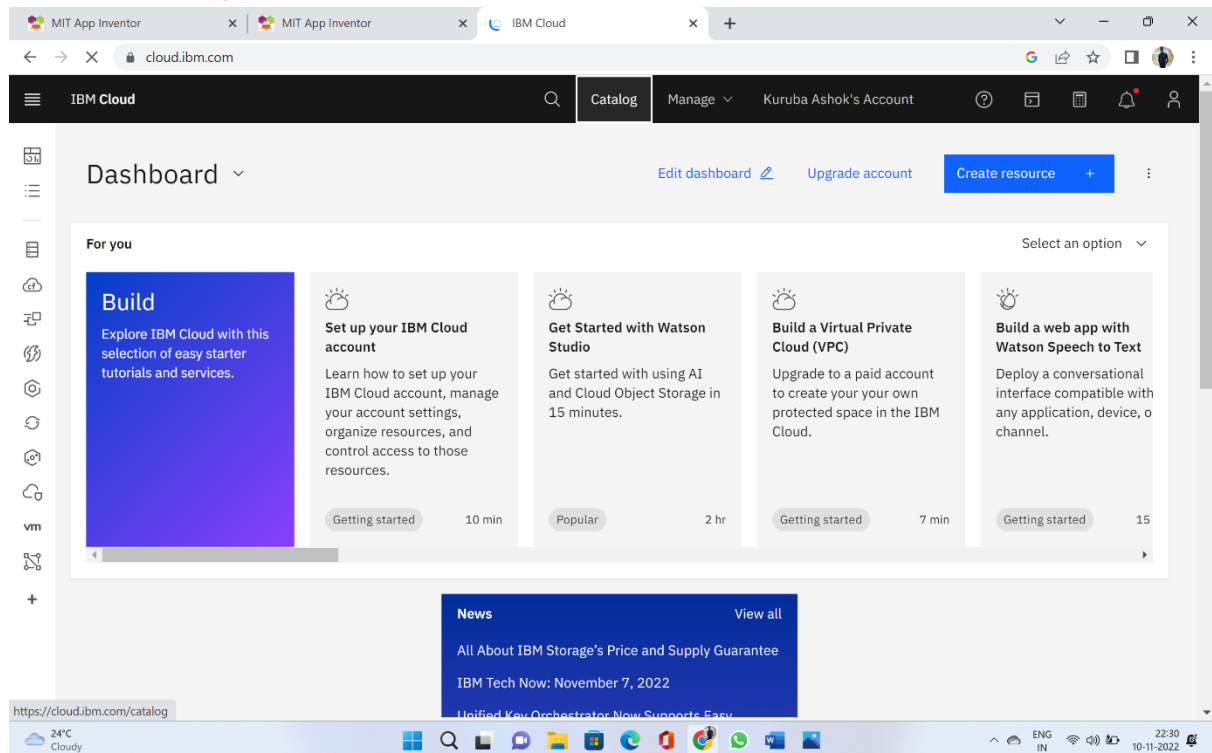
Team ID	PNT2022TMID08469
Project Name	Smart Farmer-IOT Enabled Smart Farming Application

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

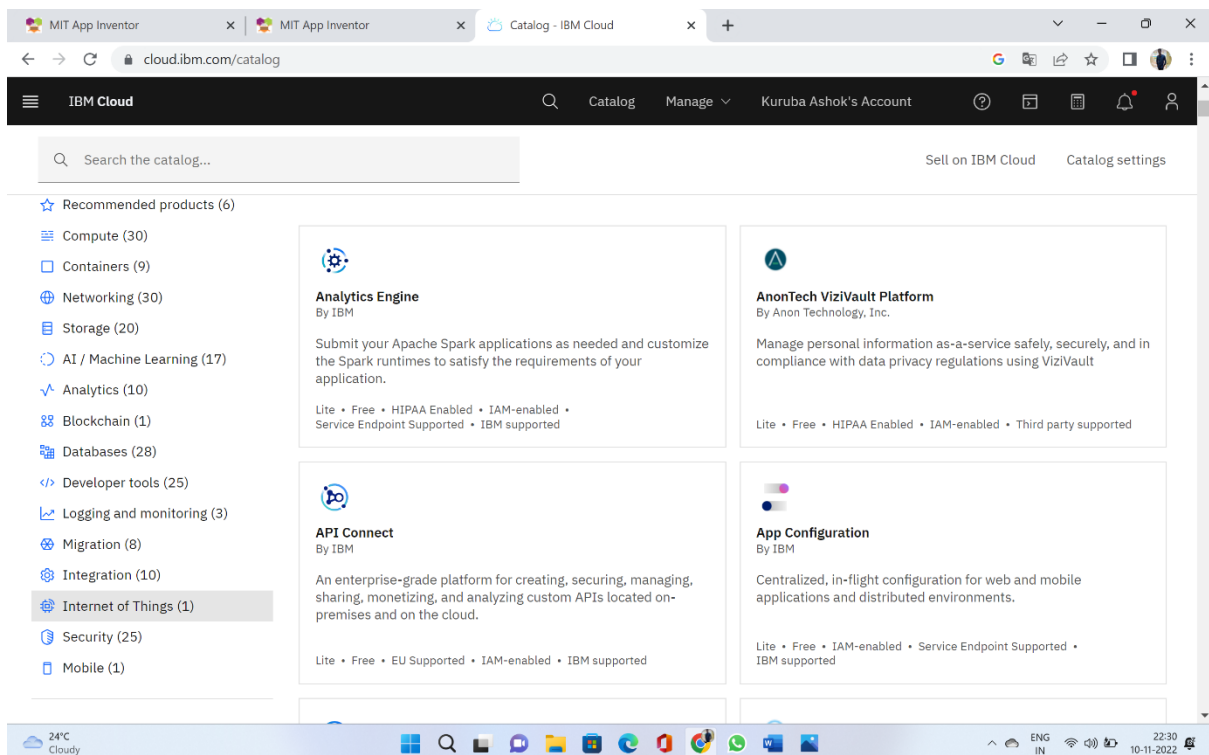
Login into IBM cloud:



Click the Catlag 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.

IBM Cloud

Internet of Things Platform

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.

Create About

Type Service

Provider IBM

Last updated 08/15/2022

Category Internet of Things

Compliance IAM-enabled

Location Frankfurt London Dallas Washington DC

Select a location

Frankfurt (eu-de)

Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or location: [United States](#)

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 Maximum of 200 MB of each of data exchanged, data analyzed and edge data analyzed	Free

Summary

Internet of Things Platform Free

Location: Frankfurt

Plan: Lite

Service name: Internet of Things Platform-gx

Resource group: Default

Existing Lite plan instance

You can have only 1 Lite plan instance of this service per resource group. [Delete](#) your current Lite plan instance in Default resource group to create a new one, or [view the existing instance](#).

☐ I have read and agree to the following license agreements: [Terms](#)

Create

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

IBM Cloud

Resource list / Internet of Things Platform-gf Active Add tags

Manage

Plan

Connections

Let's get started with IBM Watson IoT Platform

Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.

Launch Docs

Ready for the next level?

IBM Watson IoT Platform Journey

☒ Lite

The Lite service plan provides a lightweight development environment to get you started with the connectivity capabilities of Watson IoT.

☐ Non-Production

The Non-Production service plan is a full-featured, fully-integrated offering that enables you to explore Watson IoT Platform to see how

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

The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The 'Device Types' tab is active. A blue button labeled 'Add Device Type' is in the top right. Below the header, there's a section titled 'Device Types' with a description: 'This table lists all device types that are defined. You can filter the list and search for the name and description. You can modify and configure existing device types and add new device types.' A search bar is present with the placeholder 'Type the name to search ...'. Below the search bar is a table with the following data:

<input type="checkbox"/>	Name	Description	Number of Devices	Class ID	Date Added
> <input type="checkbox"/>	weather_monitor		1	Device	Nov 8, 2022 3:34 PM

Below the table, it says 'Items per page 10 | 1-1 of 1 item' and '1 of 1 page' with navigation arrows.

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

The screenshot shows the 'Add Type' window in the IBM Watson IoT Platform. The window has a progress bar at the top with two steps: 'Identity' (completed) and 'Device Information' (active). Below the progress bar, there's a section titled 'Add Type' with a description: 'These attributes will be used as a template for new devices that are assigned this device type'. An 'Edit Metadata' link is in the top right. Below the description, there are two columns of input fields:

Serial Number	Manufacturer
Enter Serial Number	Enter Manufacturer

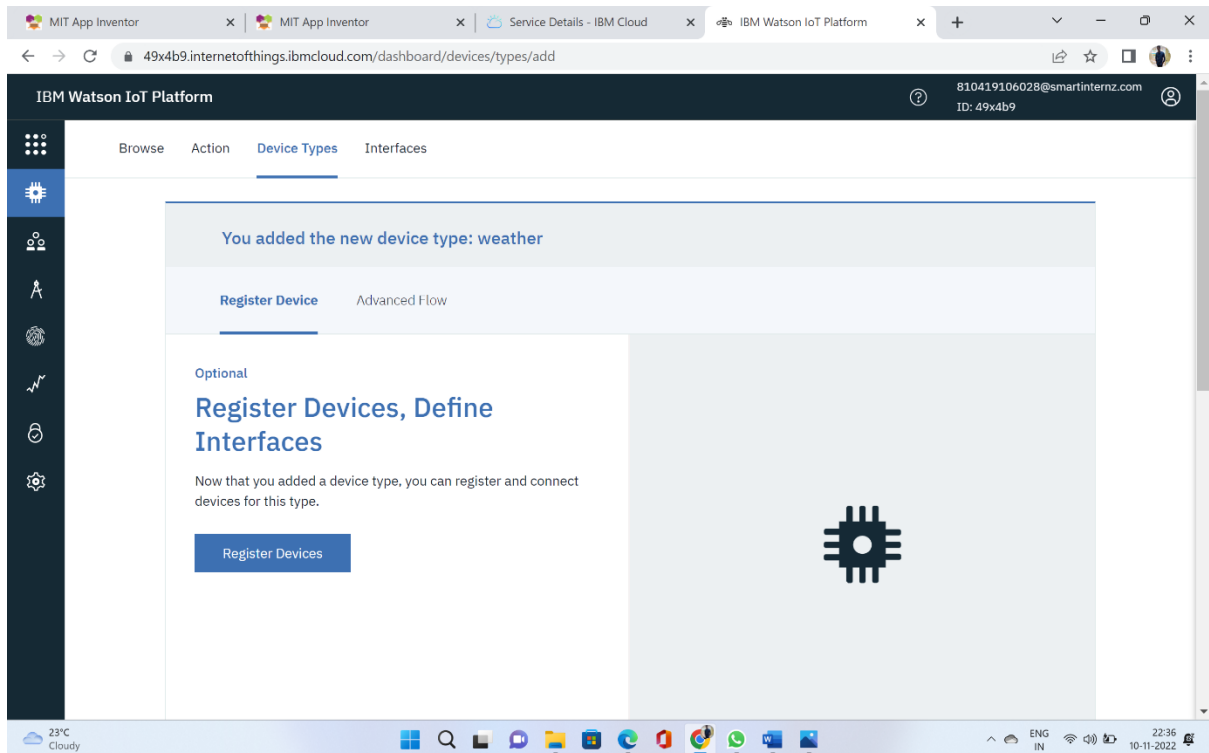
Model	Device Class
Enter Model	Enter Device Class

Description	Firmware Version
Enter Description	Enter Firmware Version

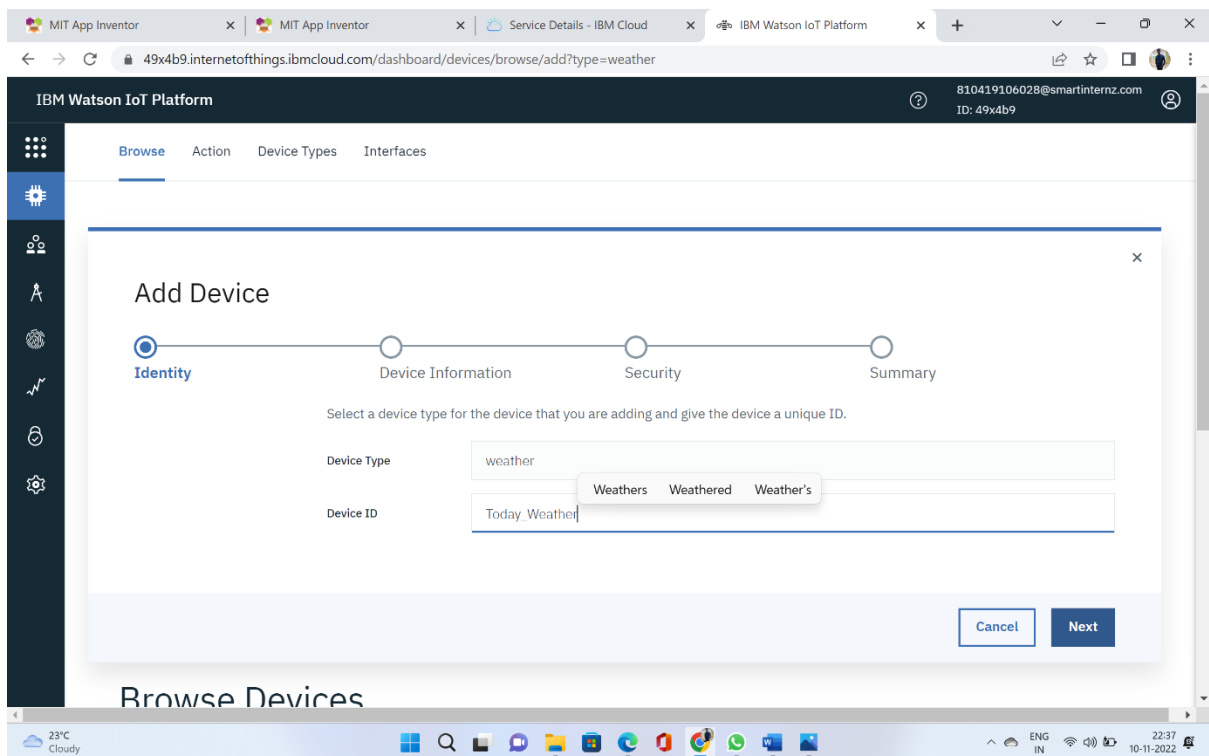
Hardware Version	Descriptive Location
Enter Hardware Version	Enter Descriptive Location

At the bottom right, there are 'Back' and 'Finish' buttons.

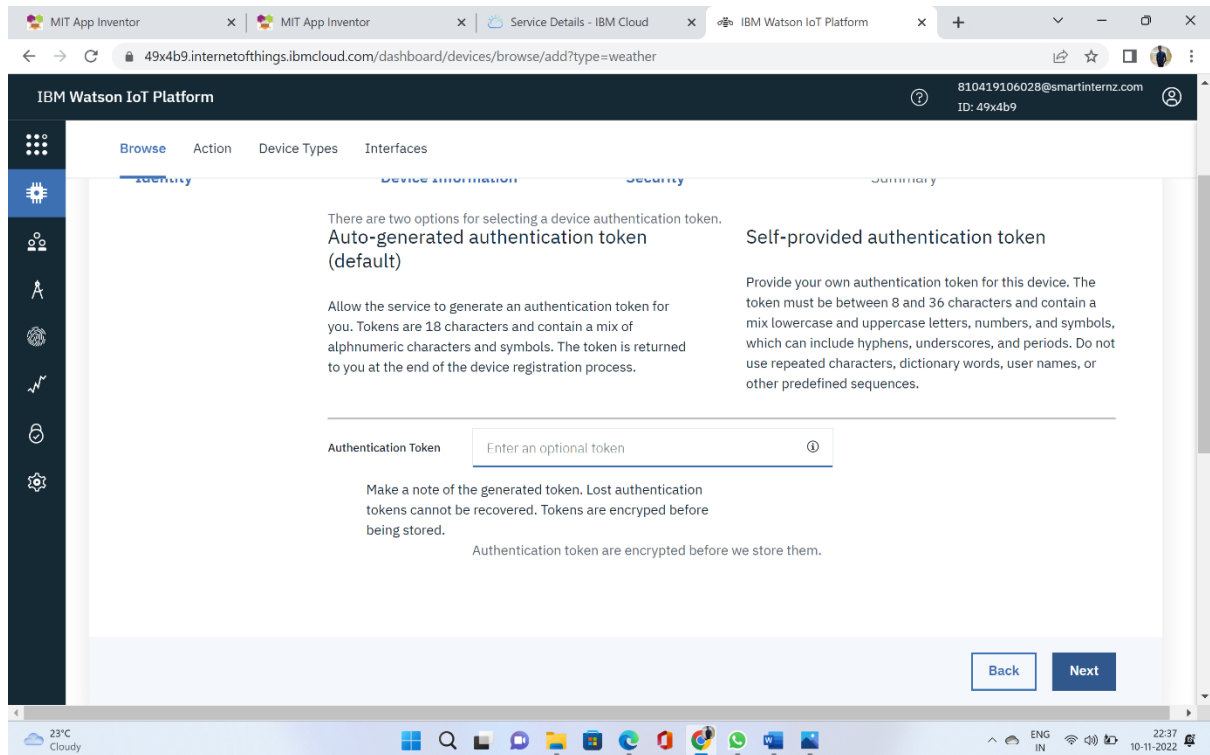
Now we have to Register Device in the Watson platform. click register device button.



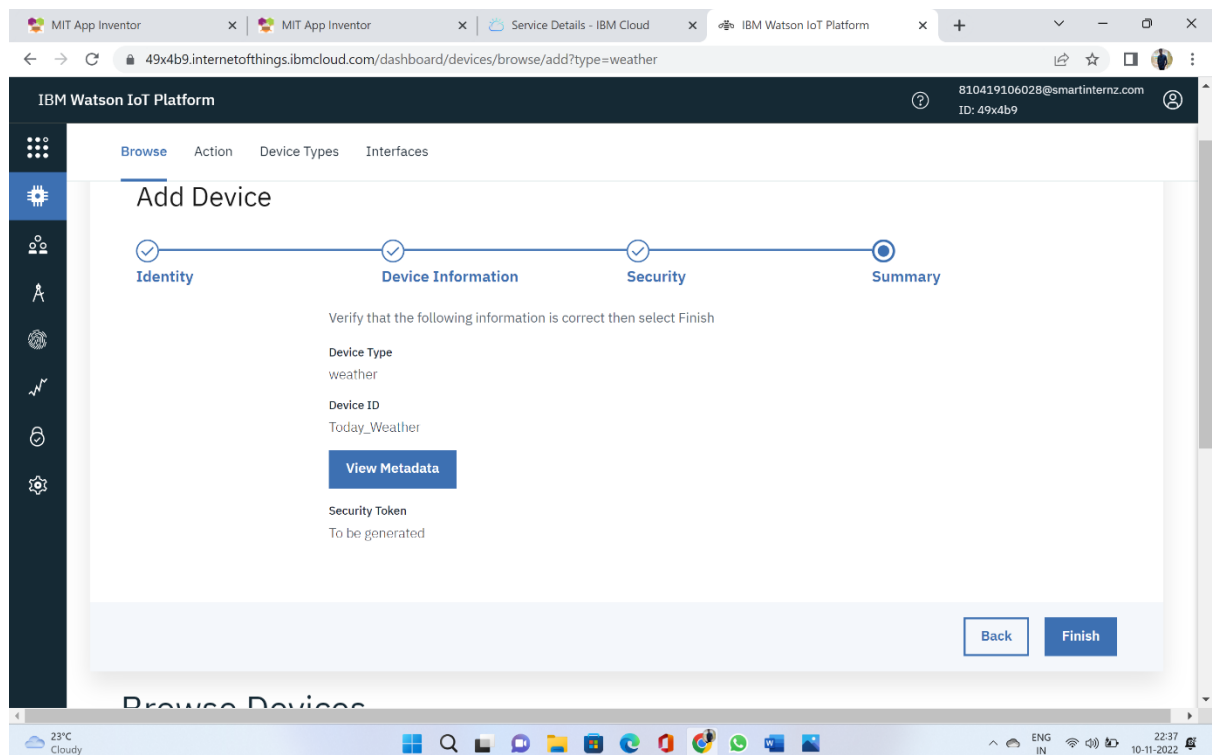
Type the device type and device id and then click the next button.



Now we have give 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 : 49x4b9

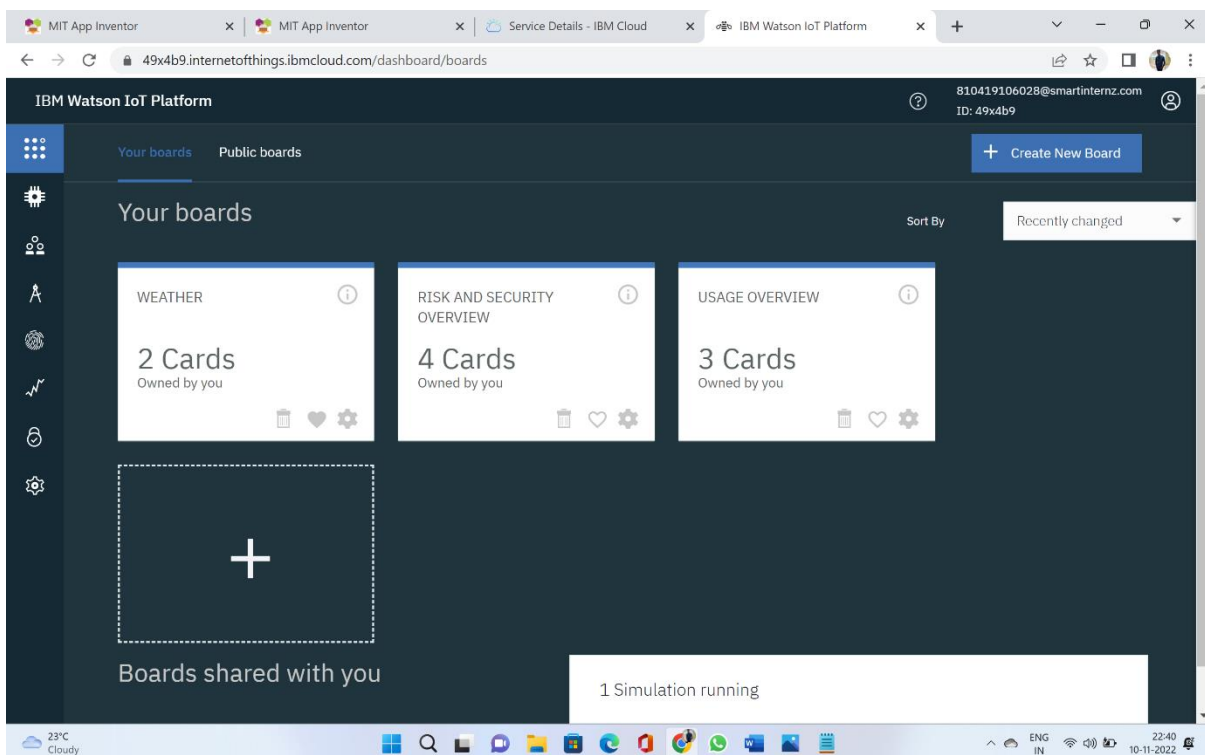
Device Type : weather_monitor

Device ID : weather_today

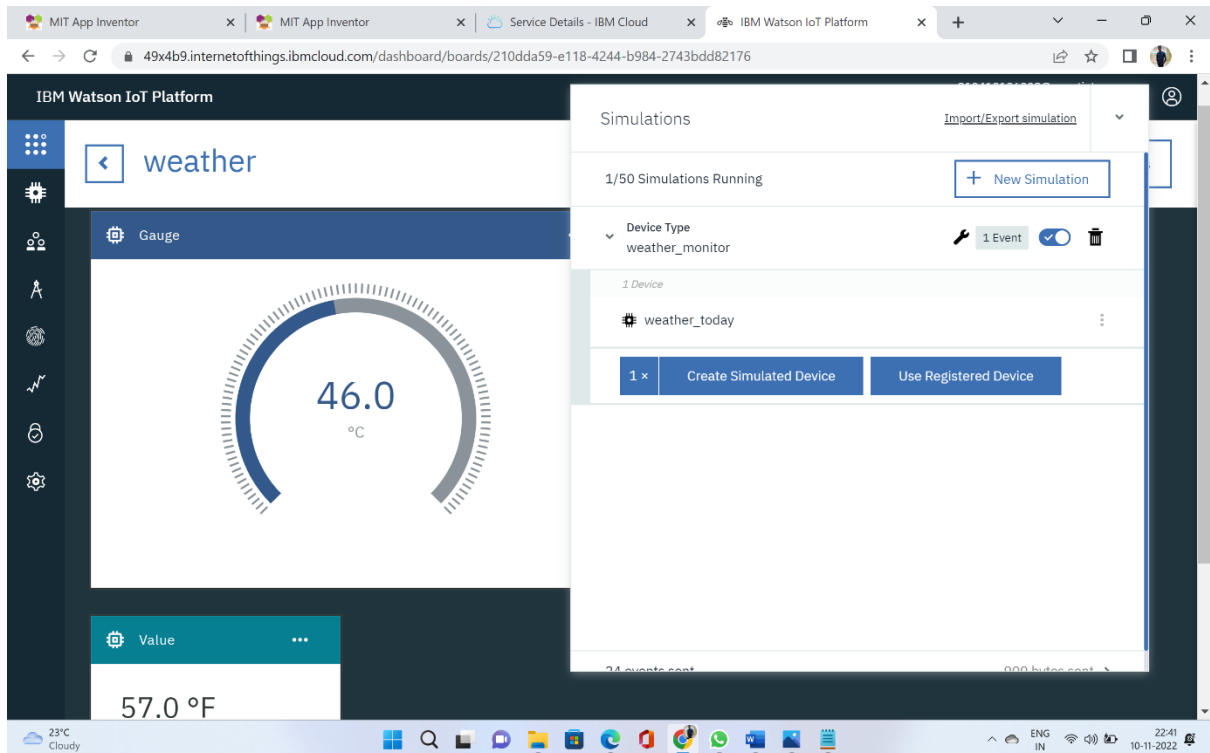
Authentication Method : use-token-auth

Authentication Token : S*xL?JyVVKPwTGH_IK

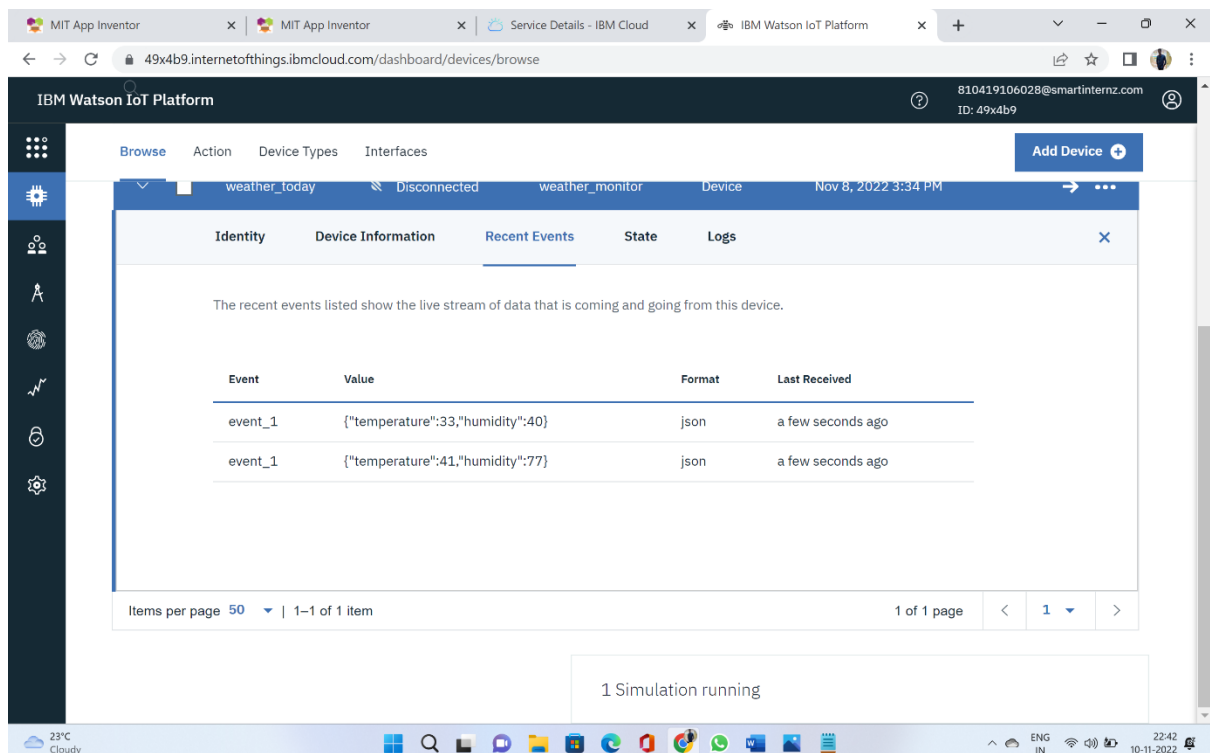
Now we have to add boards. Because we can data as graph model.



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

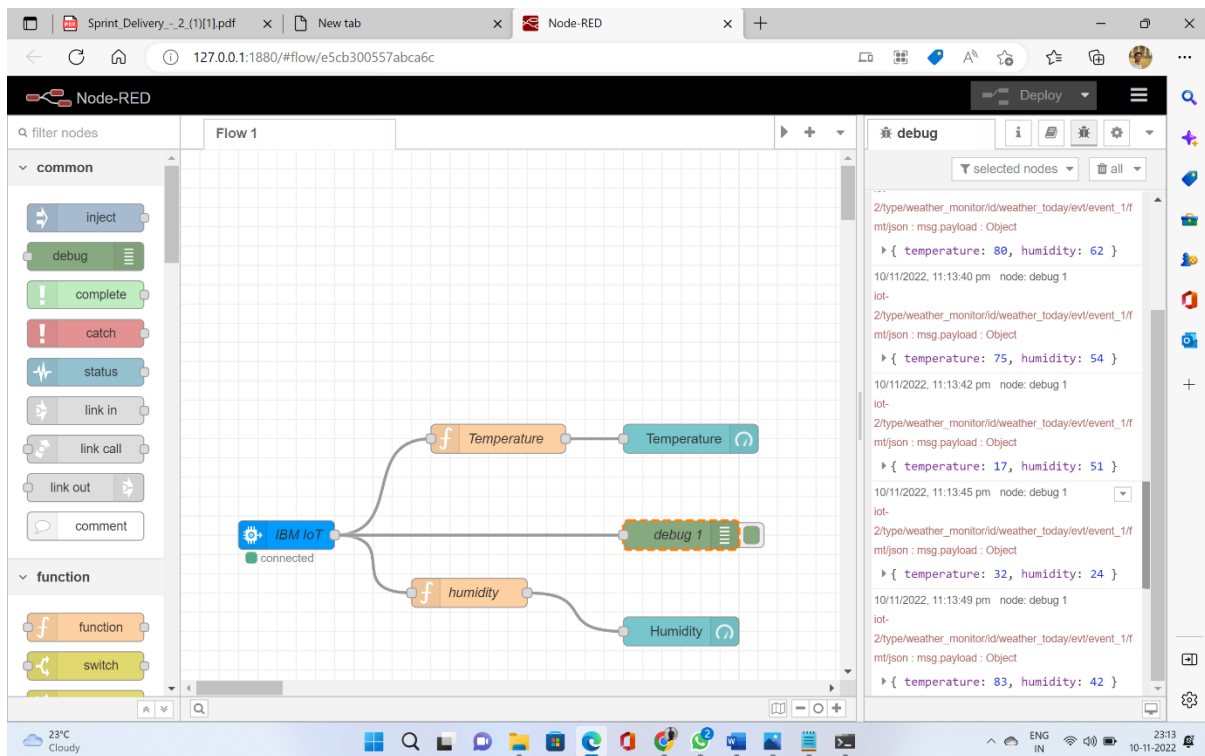


Now getting the random temperature and humidity values in the IBM Watson. You can see the received data in Recent Events under your device. So finally we can generate temperature and humidity values as like real sensors.

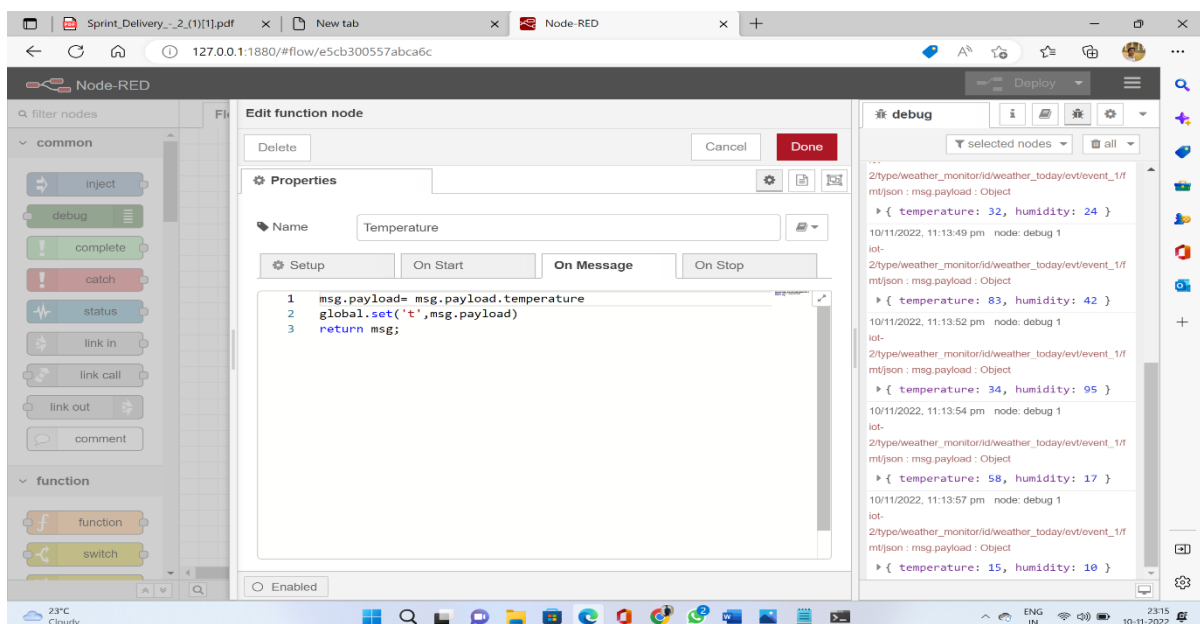


Now Configuration the Node-Red with IBM Watson Platform to collect the IBM cloud data:

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.



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



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:

The screenshot displays the Node-RED web interface in a browser. The main workspace shows a flow with a function node connected to a gauge node. The 'Edit gauge node' dialog is open, showing the following configuration:

- Group: [control] control
- Size: auto
- Type: Gauge
- Label: Temperature
- Value format: {{value}}
- Units: c
- Range: min 0, max 100
- Colour gradient: A gradient bar with green, yellow, and red segments.
- Sectors: 0, optional, optional, 100
- Class: Optional CSS class name(s) for widget

The debug console on the right shows the following messages:

```
2/type/weather_monitor/d/weather_today/evt/event_1/f
mtljson : msg.payload : Object
> { temperature: 32, humidity: 24 }
10/11/2022, 11:13:49 pm node: debug 1
iot-
2/type/weather_monitor/d/weather_today/evt/event_1/f
mtljson : msg.payload : Object
> { temperature: 83, humidity: 42 }
10/11/2022, 11:13:52 pm node: debug 1
iot-
2/type/weather_monitor/d/weather_today/evt/event_1/f
mtljson : msg.payload : Object
> { temperature: 34, humidity: 95 }
10/11/2022, 11:13:54 pm node: debug 1
iot-
2/type/weather_monitor/d/weather_today/evt/event_1/f
mtljson : msg.payload : Object
> { temperature: 58, humidity: 17 }
10/11/2022, 11:13:57 pm node: debug 1
iot-
2/type/weather_monitor/d/weather_today/evt/event_1/f
mtljson : msg.payload : Object
> { temperature: 15, humidity: 10 }
```

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

