

# SENDING DATA FROM RASPBERRY-PI TO IBM WATSON

|              |  |
|--------------|--|
| Team ID      | PNT2022TMID48300                           |
| Project Name | Gas Leakage Monitoring And Alerting System |

## **AIM:**

To send sensor data (or any dummy data) from Raspberry –Pi to IBM Watson .In our case it is DHT sensors data.

## **REQUIREMENTS:**

### **HARDWARE:**

- RASPBERRY-PI (3B)(WITH ETHERNET CABLE OR WIFI CONNECTED)
- USB MOUSE
- USB KEYBOARD
- VGA TO HDMI CABLE
- A MONITOR
- RASPBERRY'S POWER SUPPLY
- DHT-11 Sensor
- Connecting Wires

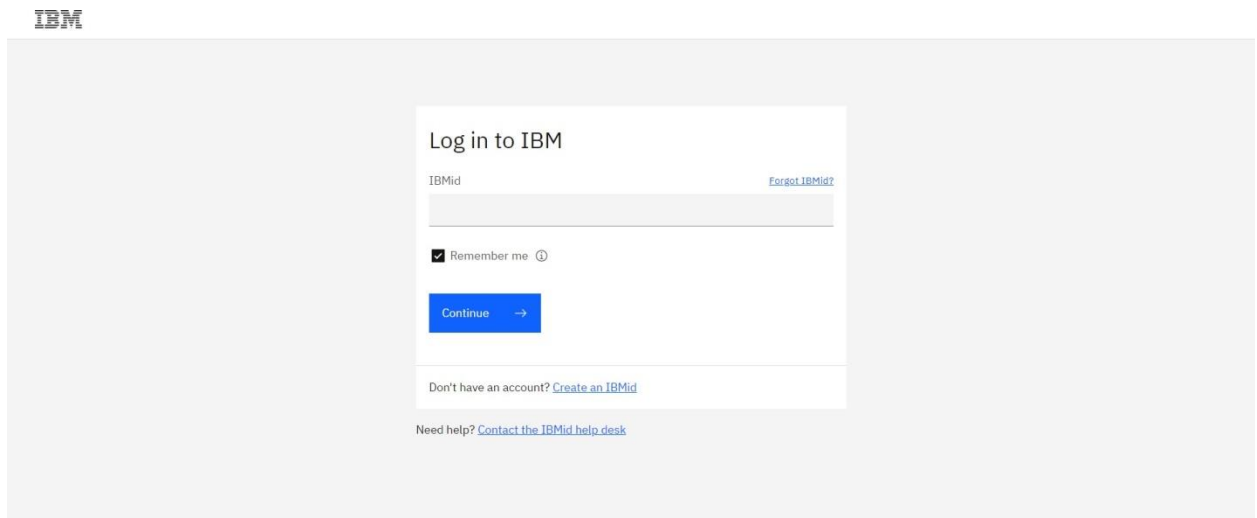
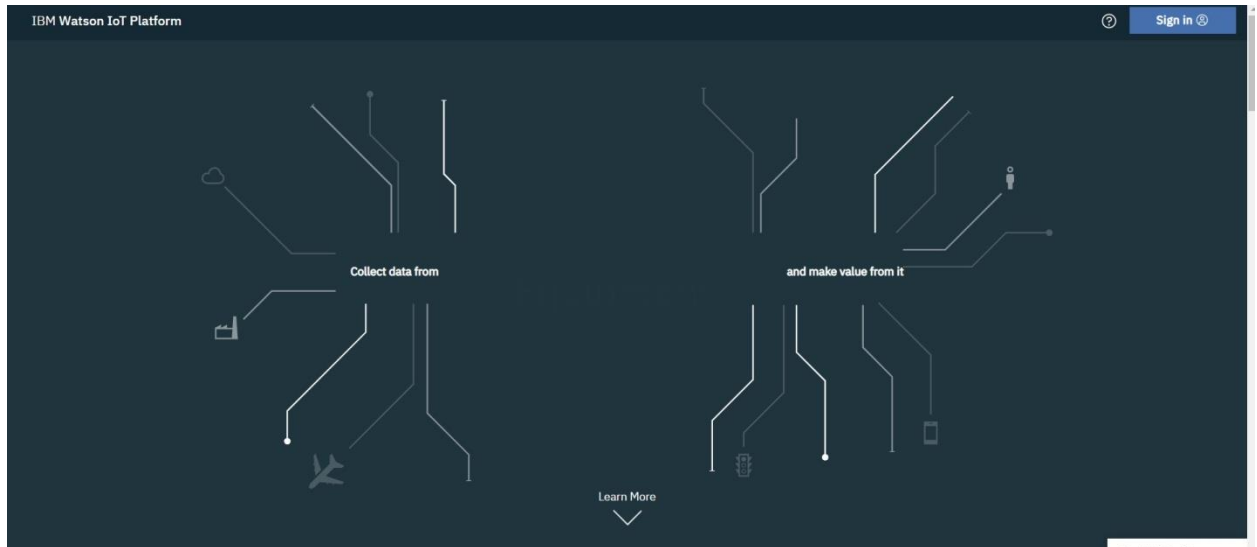
### **SOFTWARE:**

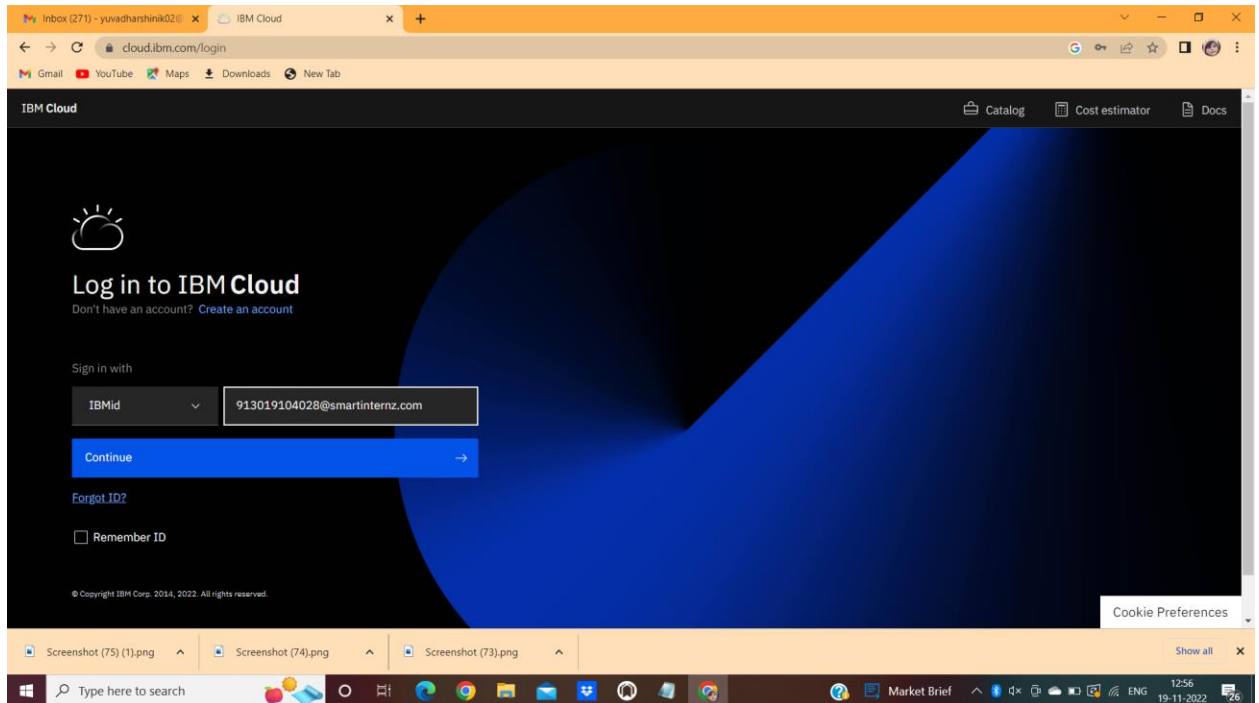
- IBM BLUEMIX ACCOUNT

## STEPS TO BE FOLLOWED:

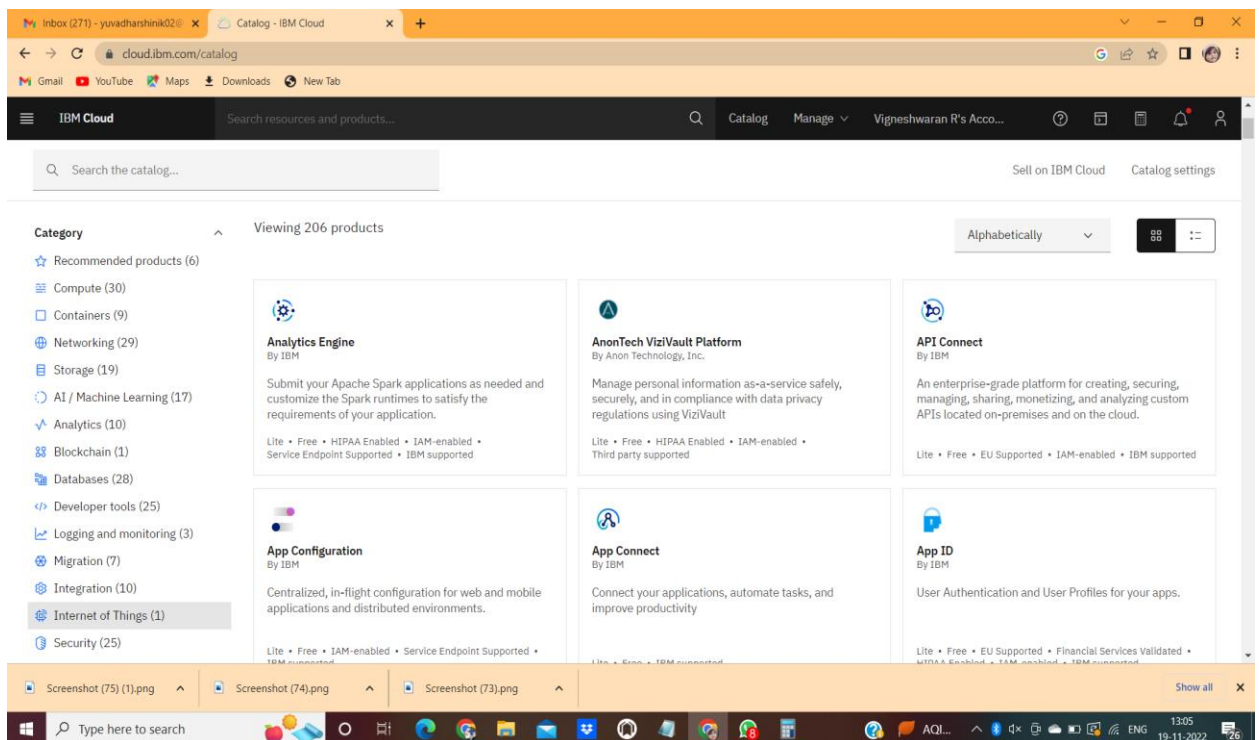
### **STEP 1:** Create a device in IBM Watson

➤ Firstly, login into your IBM-Bluemix account with your e-mail ID and Password.





➤ Click on catalog on your dashboard screen, then under platform go IoT.



➤ Check all details and click on create.

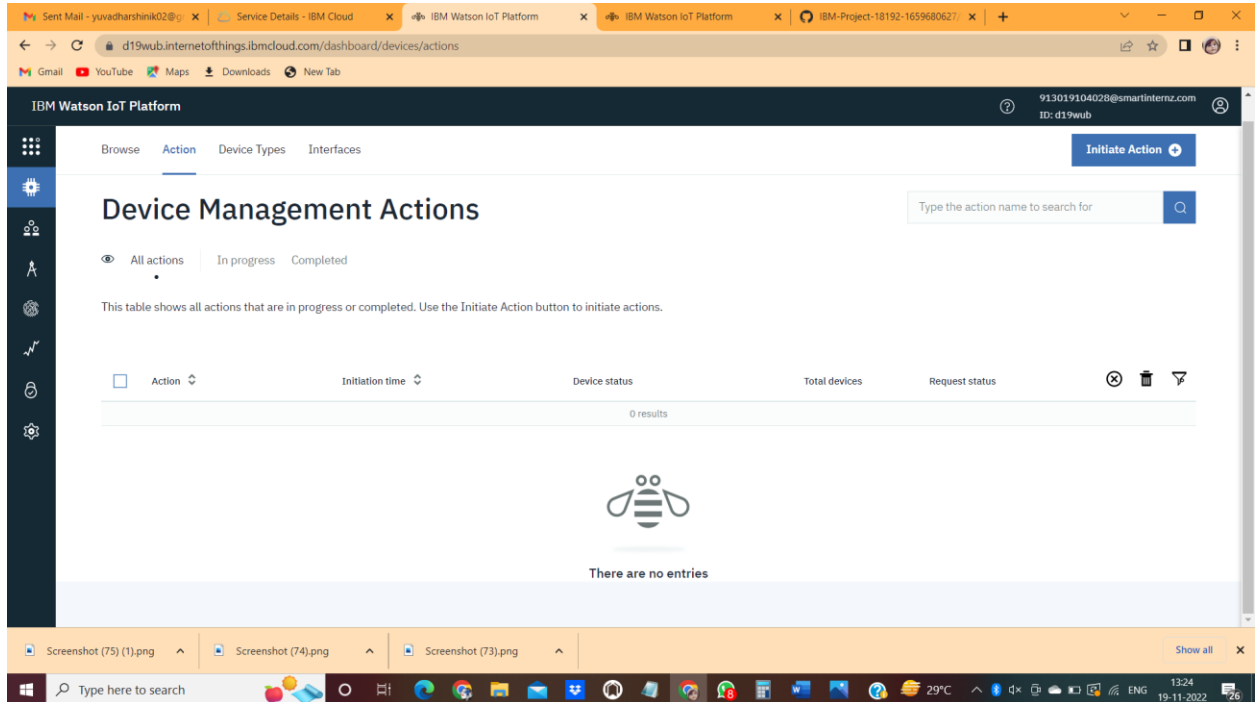
The screenshot shows the IBM Cloud catalog page for the Internet of Things Platform. The page is titled "Internet of Things Platform" and describes it as the hub for all things IBM IoT. It features a "Create" button and a "Summary" sidebar. The "Summary" sidebar includes details such as Location (London), Plan (Lite), Service name (Internet of Things Platform-lh), and Resource group (Default). A checkbox indicates that the user has read and agreed to the license agreements. A "Create" button is prominently displayed. The main content area shows a "Select a location" dropdown menu set to "London (eu-gb)" and a "Select a pricing plan" section. The pricing table shows a "Lite" plan for free, which includes up to 500 registered devices and a maximum of 200 MB of each data metric. The table also lists features like maximum registered devices, application bindings, and data exchange/analysis limits.

| Plan | Features  | Pricing |
|------|---|---------|
| Lite | Includes up to 500 registered devices, and a maximum of 200 MB of each data metric<br>Maximum of 500 registered devices<br>Maximum of 500 application bindings<br>Maximum of 200 MB of each of data exchanged, data analyzed and edge data analyzed | Free    |

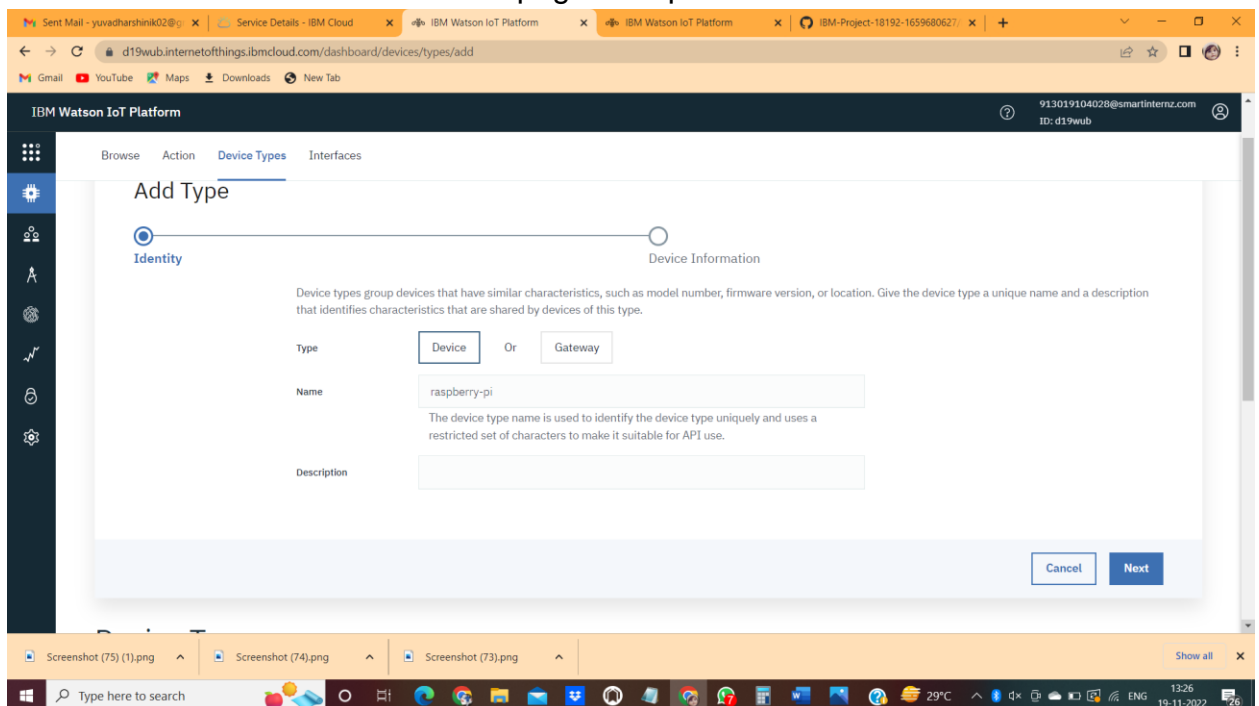
➤ Click on Launch

The screenshot shows the IBM Cloud "Service Details" page for the Internet of Things Platform. The page is titled "Internet of Things Platform-1x" and is marked as "Active". It features a "Launch" button and a "Docs" button. The main content area includes a diagram of a device connected to a cloud, and a section titled "Let's get started with IBM Watson IoT Platform" with the text "Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world." Below this, there is a "Ready for the next level?" section with a "IBM Watson IoT Platform Journey" diagram. The diagram shows three stages: "Lite", "Non-Production", and "Production". The "Lite" stage is highlighted, and its description is: "The Lite service plan provides a lightweight development environment to get you started". The "Non-Production" stage is described as "The Non-Production service plan is a full-featured, fully-integrated offering that enables". The "Production" stage is described as "The Production service is a fully managed SaaS offering that enables you to manage and analyze".

➤ Dashboard of IBM Watson IoT platform, Click on Add device.



➤ After click on Add device this page will open



➤ Go to device type and fill the details.

The screenshot shows the 'Add Type' form in the IBM Watson IoT Platform. The form is titled 'Add Type' and has two steps: 'Identity' (current step) and 'Device Information'. The 'Identity' step includes a progress bar with a blue circle indicating the current step. Below the progress bar, there is a text box for 'Name' containing 'raspberry-pi' and a text box for 'Description'. The 'Device Information' step is currently inactive. At the bottom right, there are 'Cancel' and 'Next' buttons.

IBM Watson IoT Platform

Browse Action **Device Types** Interfaces

### Add Type

Identity Device Information

Device types group devices that have similar characteristics, such as model number, firmware version, or location. Give the device type a unique name and a description that identifies characteristics that are shared by devices of this type.

Type  Or

Name

The device type name is used to identify the device type uniquely and uses a restricted set of characters to make it suitable for API use.

Description

➤ Click on Finish

The screenshot shows the 'Add Type' form in the IBM Watson IoT Platform, Step 2: Device Information. The 'Identity' step is now complete, indicated by a blue checkmark. The 'Device Information' step is active, showing a progress bar with a blue circle. Below the progress bar, there is a text box for 'Name' containing 'raspberry-pi' and a text box for 'Description'. The 'Device Information' step includes a text box for 'Serial Number' and a text box for 'Manufacturer'. Below these, there are text boxes for 'Model', 'Device Class', 'Description', 'Firmware Version', 'Hardware Version', and 'Descriptive Location'. At the bottom right, there are 'Back' and 'Finish' buttons.

IBM Watson IoT Platform

Browse Action **Device Types** Interfaces

### Add Type

Identity Device Information

These attributes will be used as a template for new devices that are assigned this device type

Serial Number

Manufacturer

Model

Device Class

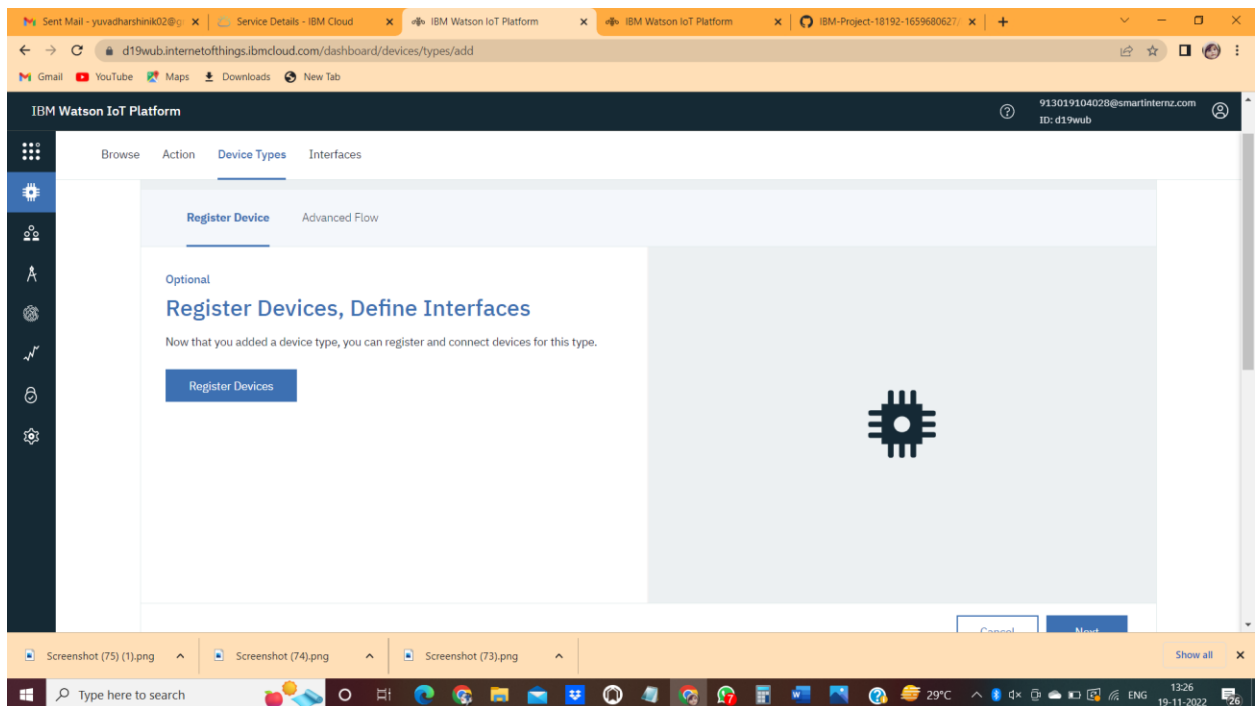
Description

Firmware Version

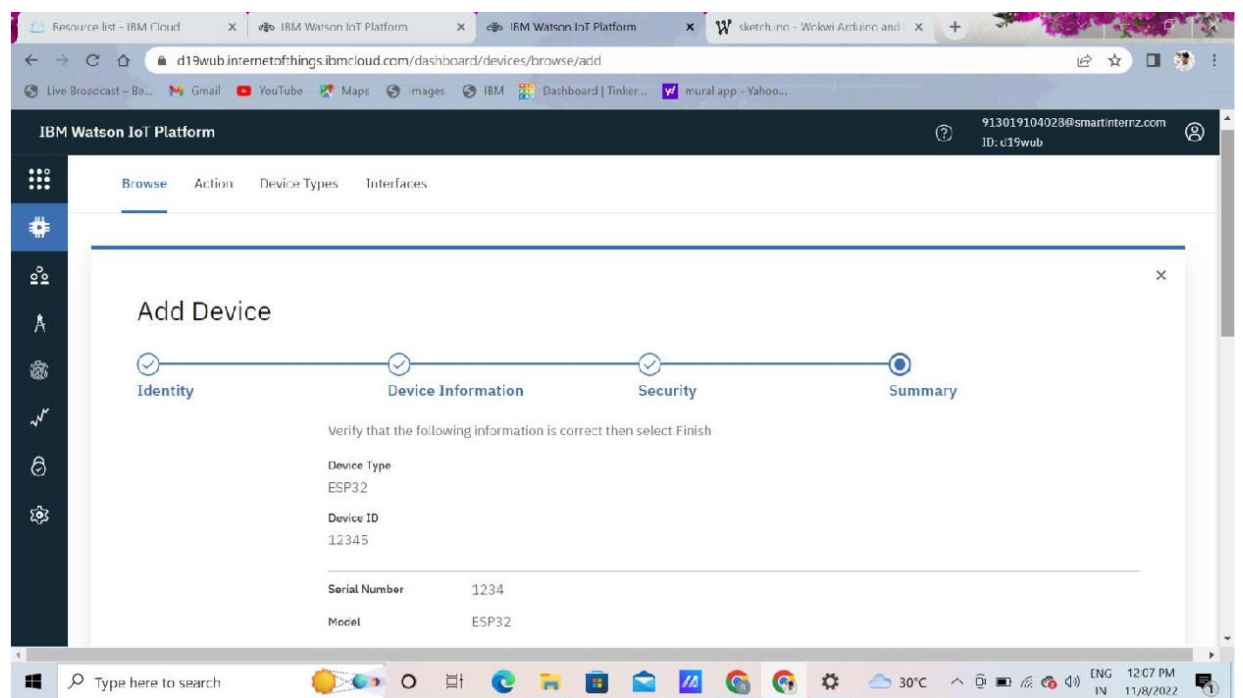
Hardware Version

Descriptive Location

➤ Click on Register Device.



➤ Choose the device and give device ID and then click on next.



➤ Device is created.

The screenshot displays the IBM Watson IoT Platform interface. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains icons for various platform features. The main content area is titled 'Browse Devices' and includes a search bar and a table of devices. The table shows one device with ID 1234, status 'Connected', and type 'raspberrypi'. A 'Device Simulator' toggle is visible on the right. The bottom of the screen shows a Windows taskbar with various application icons and system status indicators.

IBM Watson IoT Platform

913019104028@smartinternz.com  
ID: d19wub

[Add Device](#)

## Browse Devices

[All Devices](#) [Diagnose](#)

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator ☒

| <input type="checkbox"/>   | Device ID | Status   | Device Type | Class ID | Date Added            | Descriptive Location |
|----------------------------|-----------|--|-------------|----------|-----------------------|----------------------|
| > <input type="checkbox"/> | 1234      | <span style="color: green;">●</span> Connected | raspberrypi | Device   | Nov 18, 2022 11:14 PM |                      |

Items per page 50 | 1-1 of 1 item

1 of 1 page

0 Simulations running

ENG IN 00:01 19-11-2022



## STEP-2: INSTALLING NECESSARY PACKAGES ON YOUR PI:

- Now we are going to install necessary packages on your pi.
- Open your terminal in your pi and type the following commands
- `curl -LO https://github.com/ibm-messaging/iot-raspberrypi/releases/download/1.0.2.1/iot_1.0-2_armhf.deb`
- `sudo dpkg -i iot_1.0-2_armhf.deb`
- `service iot status`

Following are the images as to what appears on your pi's terminal when u type these commands:

```
File Edit Tabs Help
--2017-10-23 06:55:22-- http://ftp.nl.debian.org/debian/pool/main/o/openssl/lib
ssl1.0.0_1.0.1t-1+deb8u6_armhf.deb
Resolving ftp.nl.debian.org (ftp.nl.debian.org)... 130.89.149.21, 2001:67c:2564:
a120::21
Connecting to ftp.nl.debian.org (ftp.nl.debian.org)|130.89.149.21|:80... connect
ed.
HTTP request sent, awaiting response... 200 OK
Length: 867950 (848K) [application/x-debian-package]
Saving to: 'libssl1.0.0_1.0.1t-1+deb8u6_armhf.deb'

libssl1.0.0_1.0.1t- 100%[=====] 847.61K  358KB/s   in 2.4s

2017-10-23 06:55:25 (358 KB/s) - 'libssl1.0.0_1.0.1t-1+deb8u6_armhf.deb' saved [
867950/867950]

pi@raspberrypi:~$ sudo dpkg -i libssl1.0.0_1.0.1t-1+deb8u6_armhf.deb
Selecting previously unselected package libssl1.0.0:armhf.
(Reading database ... 115606 files and directories currently installed.)
Preparing to unpack libssl1.0.0_1.0.1t-1+deb8u6_armhf.deb ...
Unpacking libssl1.0.0:armhf (1.0.1t-1+deb8u6) ...
Setting up libssl1.0.0:armhf (1.0.1t-1+deb8u6) ...
pi@raspberrypi:~$ curl -LO https://github.com/ibm-messaging/iot-raspberrypi/rel
eases/download/1.0.2.1/iot_1.0-2_armhf.deb
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Done    0     0    0      0      0      0      0      0
100 164    0 164    0    157    0  --:--:--  0:00:01 --:--:-- 157
100 609    0 609    0    457    0  --:--:--  0:00:01 --:--:-- 457
100 110k 100 110k    0    29117    0  0:00:03  0:00:03 --:--:-- 48190
pi@raspberrypi:~$ sudo dpkg -i iot_1.0-2_armhf.deb
(Reading database ... 115626 files and directories currently installed.)
Preparing to unpack iot_1.0-2_armhf.deb ...
Unpacking iot (1.0-1) over (1.0-1) ...
Setting up iot (1.0-1) ...
Processing triggers for systemd (232-25+deb9u1) ...
pi@raspberrypi:~$ service iot status
● iot.service - LSB: IoT service
   loaded: loaded (/etc/init.d/iot; generated; vendor preset: enabled)
   Active: active (running) since Mon 2017-10-23 06:56:25 UTC; 17s ago
     Docs: man:systemd-sysv-generator(8)
    Group: /system.slice/iot.service
           └─2562 /opt/iot/iot /dev/null

Oct 23 06:56:24 raspberrypi systemd[1]: Starting LSB: IoT service...
Oct 23 06:56:24 raspberrypi iot[2562]: Starting the iot program
Oct 23 06:56:25 raspberrypi iot[2562]: **** IoT Raspberry Pi Sample has started ****
Oct 23 06:56:25 raspberrypi iot[2562]: Config file not found. Going to Quickstart mode
Oct 23 06:56:25 raspberrypi iot[2562]: Running in Quickstart mode
Oct 23 06:56:25 raspberrypi systemd[1]: Started LSB: IoT service.
```

➤ Then open your terminal and type pip install ibmiotf

```
File Edit Tabs Help
pi@raspberrypi:~$ pip install ibmiotf
Collecting ibmiotf
  Downloading ibmiotf-0.3.0.tar.gz (58kB)
    100% |#####| 61kB 510kB/s
Collecting dicttoxml>=1.7.4 (from ibmiotf)
  Downloading dicttoxml-1.7.4.tar.gz
Collecting iso8601>=0.1.10 (from ibmiotf)
  Downloading iso8601-0.1.12-py2.py3-none-any.whl
Collecting paho-mqtt>=1.2 (from ibmiotf)
  Downloading paho-mqtt-1.3.1.tar.gz (88kB)
    100% |#####| 81kB 916kB/s
Collecting pytz>=2014.7 (from ibmiotf)
  Using cached pytz-2017.2-py2.py3-none-any.whl
Collecting requests>=2.5.0 (from ibmiotf)
  Downloading requests-2.18.4-py2.py3-none-any.whl (88kB)
    100% |#####| 92kB 1.6MB/s
Collecting requests-toolbelt>=0.9.0 (from ibmiotf)
  Downloading requests-toolbelt-0.8.0-py2.py3-none-any.whl (54kB)
    100% |#####| 61kB 1.6MB/s
Collecting xmldict>=0.10.2 (from ibmiotf)
  Downloading xmldict-0.11.8-py2.py3-none-any.whl
Collecting urllib3<1.23,>=1.21.1 (from requests>=2.5.0->ibmiotf)
  Downloading urllib3-1.22-py2.py3-none-any.whl (124kB)
    100% |#####| 133kB 1.4MB/s
Collecting idna<2.7,>=2.5 (from requests>=2.5.0->ibmiotf)
  Downloading idna-2.6-py2.py3-none-any.whl (56kB)
    100% |#####| 61kB 1.7MB/s
Collecting chardet<3.1.0,>=3.0.2 (from requests>=2.5.0->ibmiotf)
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133kB)
    100% |#####| 143kB 1.0MB/s
Collecting certifi>=2017.4.17 (from requests>=2.5.0->ibmiotf)
  Using cached certifi-2017.7.27.1-py2.py3-none-any.whl
Building wheels for collected packages: ibmiotf, dicttoxml, paho-mqtt
  Running setup.py bdist_wheel for ibmiotf ... done
  Stored in directory: /home/pi/.cache/pip/wheels/7e/fa/45/bbc33ad957e82f7b71ba80e316d65a83d9d735a0d12e0c0418
  Running setup.py bdist_wheel for dicttoxml ... done
  Stored in directory: /home/pi/.cache/pip/wheels/45/62/59/96910b33ec6a7b2ae66a13765401b50def5468024078e12cce
  Running setup.py bdist_wheel for paho-mqtt ... done
  Stored in directory: /home/pi/.cache/pip/wheels/20/d8/0d/acdc0f2890111b7be7de71deebef0e42fb83be0313dffe493
Successfully built ibmiotf dicttoxml paho-mqtt
Installing collected packages: dicttoxml, iso8601, paho-mqtt, pytz, urllib3, idna, chardet, certifi, requests, requests-toolbelt, xmldict, ibmiotf
Successfully installed certifi-2017.7.27.1 chardet-3.0.4 dicttoxml-1.7.4 ibmiotf-0.3.0 idna-2.6 iso8601-0.1.12 paho-mqtt-1.3.1 pytz-2017.2 requests-2.18.4 requests-toolbelt-0.8.0 urllib3-1.22 xmldict-0.11.0
pi@raspberrypi:~$
```

➤ I have sent DHT-11 Sensors data to ibm bluemix .To get the code u need to login into IOT GYAN.

### STEP-3: Checking your data sent on IBM Bluemix:

➤ After you have sent your sensors data you can check whether it is received at your iot platform Just look at the image below and if u see the same wifi kind of symbol on your created device then your data is being received.

IBM Watson IoT Platform

Browse Action Device Types Interfaces

## Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, organized, and searched on using different criteria. To get started, you can add devices by using the Add Device button, or by using API.

Search by Device ID

Device Simulator ☒

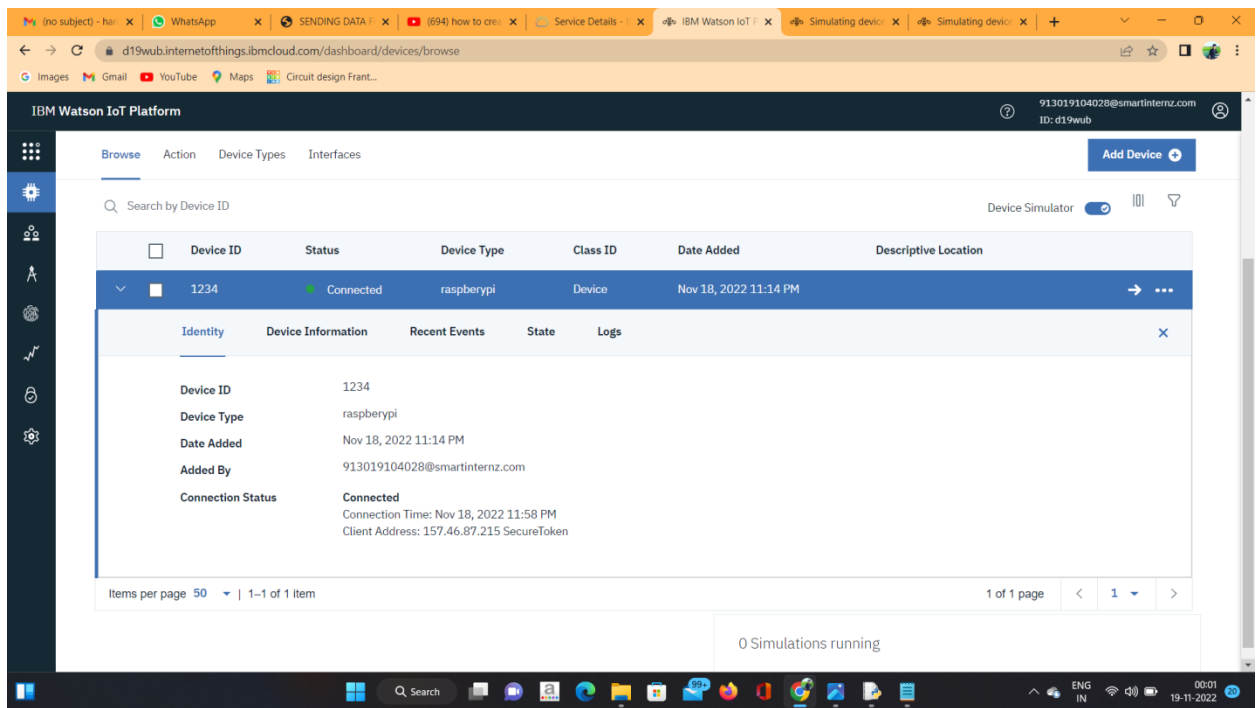
| Device ID | Status    | Device Type | Class ID | Date Added            | Descriptive Location |
|-----------|-----------|-------------|----------|-----------------------|----------------------|
| 1234      | Connected | raspberrypi | Device   | Nov 18, 2022 11:14 PM |                      |

Items per page 50 | 1-1 of 1 item

1 of 1 page

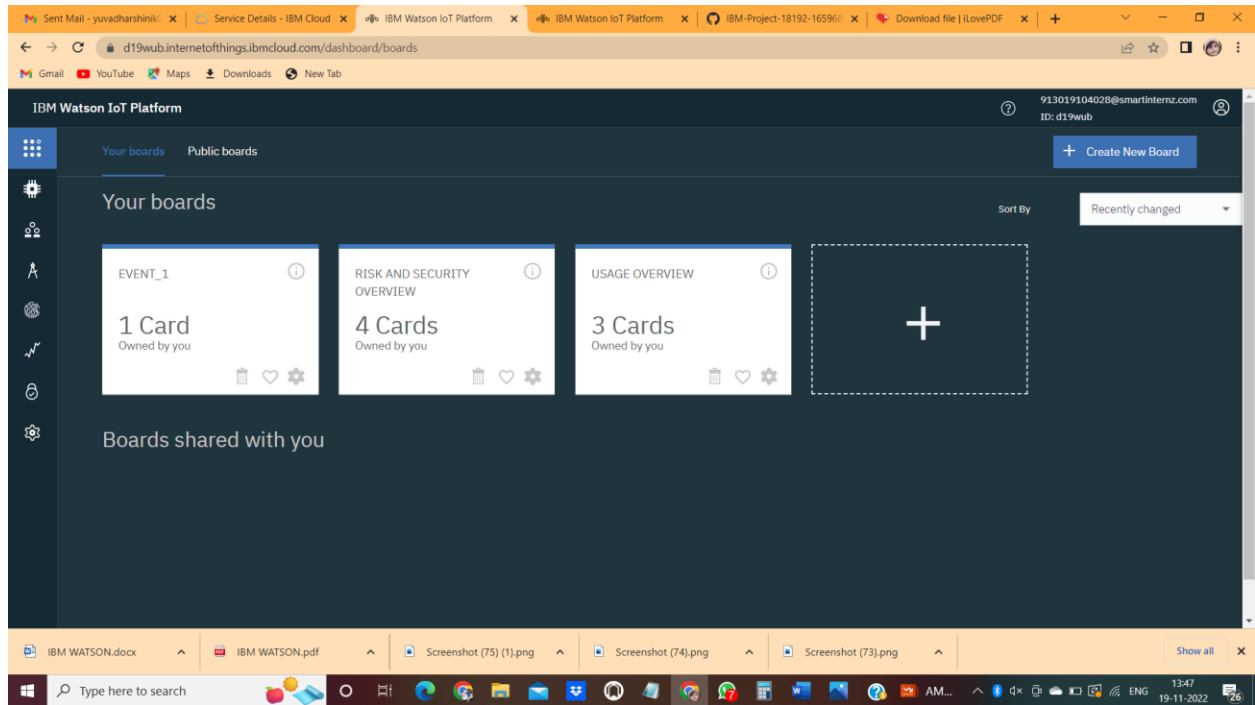
0 Simulations running

➤ After double clicking on your created device you can see the received data as shown in image.

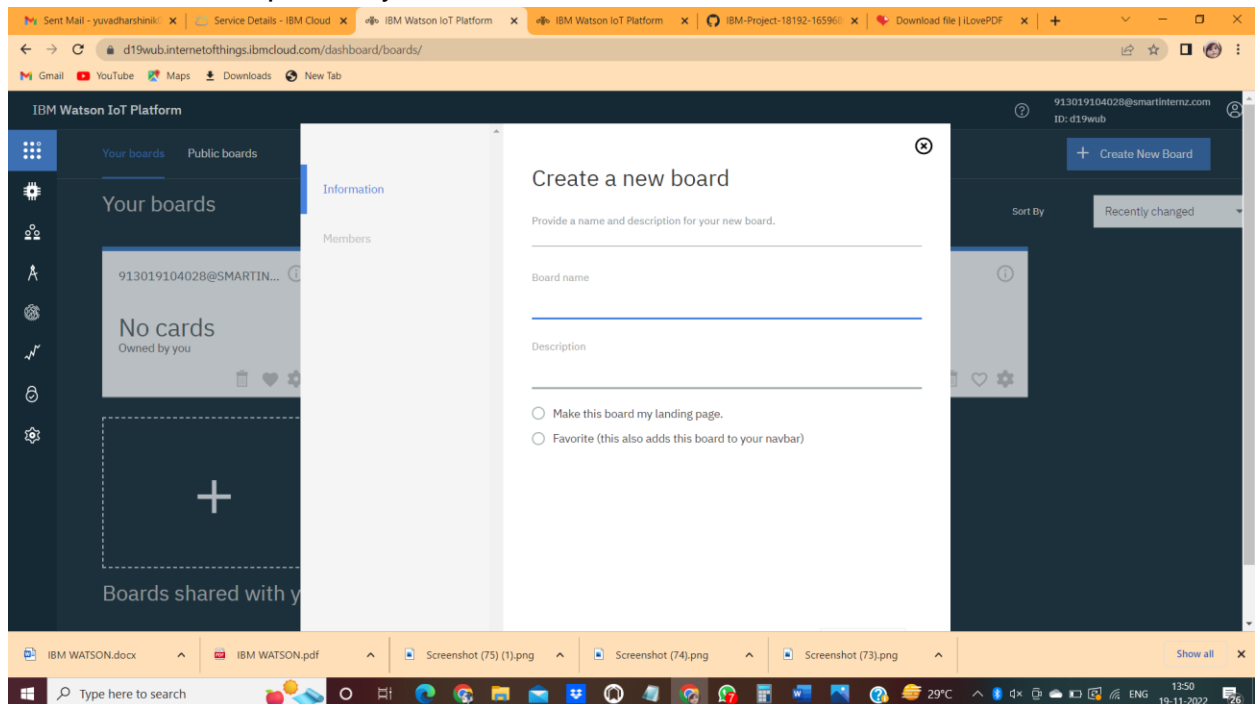


#### STEP-4: Creating boards and cards for visualization of data:

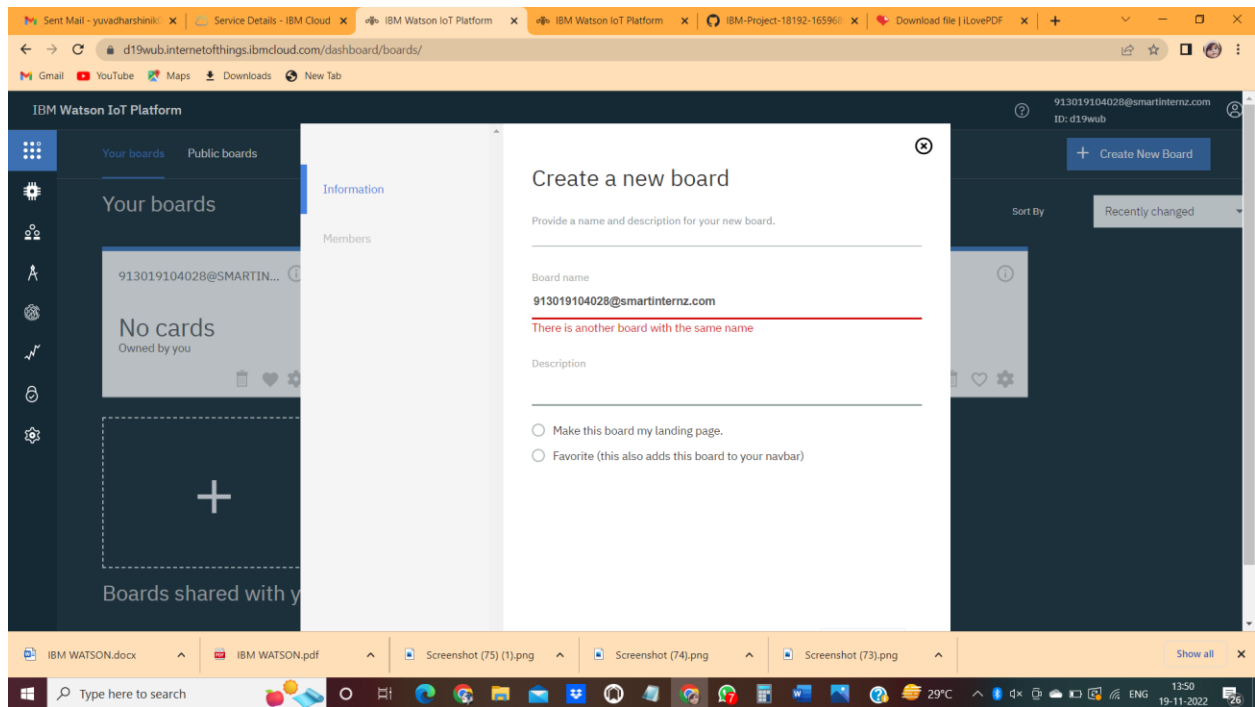
➤ In your Watson platform you have an option called board .Click on it and you get the following window on your screen.



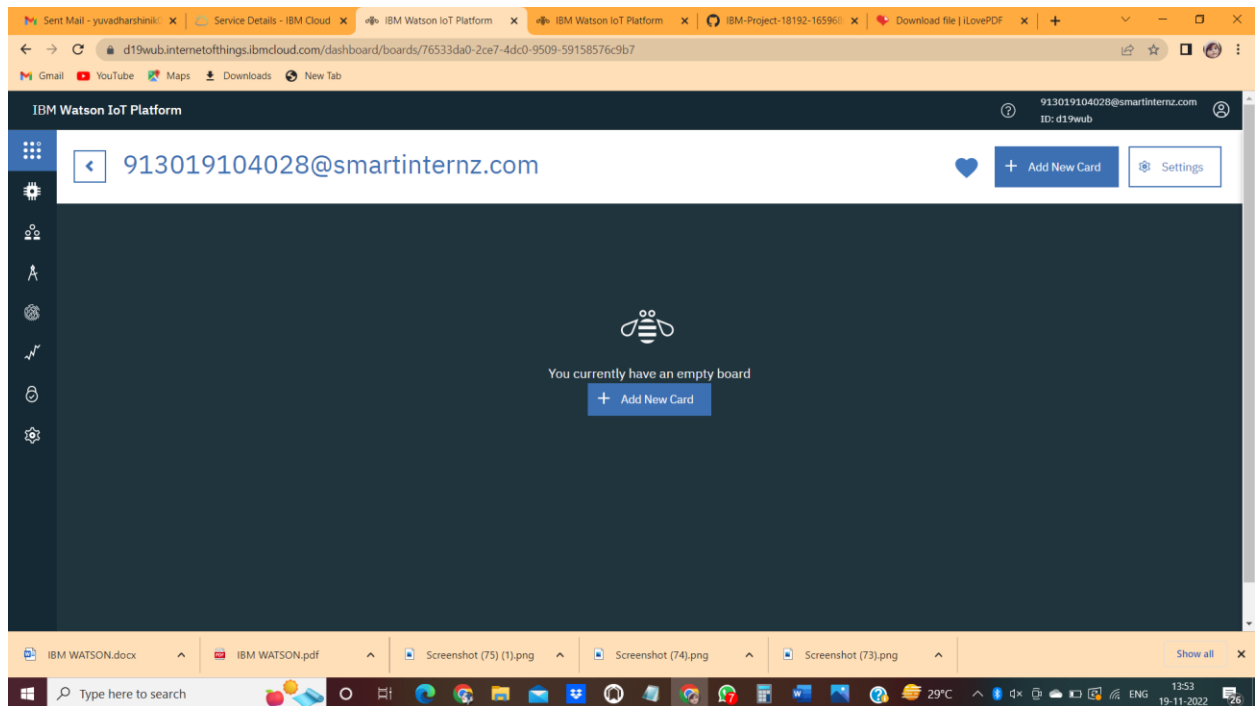
➤ Click on Create a new board to create a board. The given below window appears give a name and description to your board as shown in the window below.



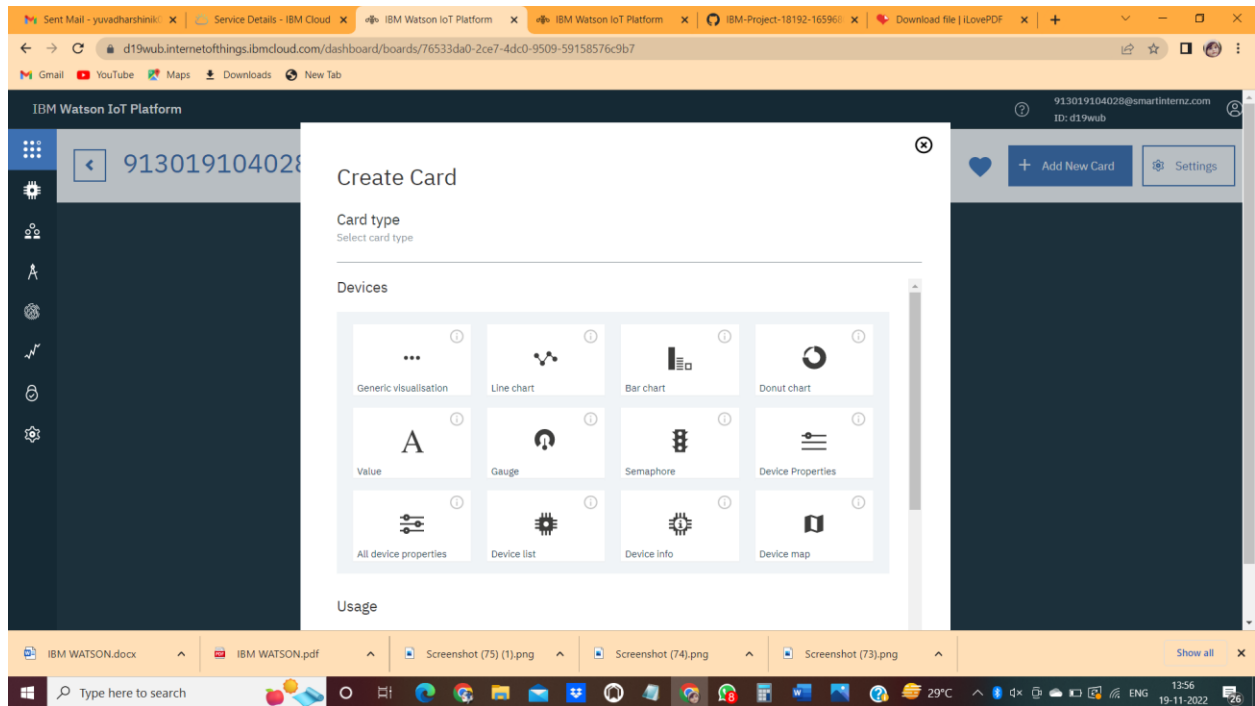
➤ Then click on Next you get the below window then again click on Submit



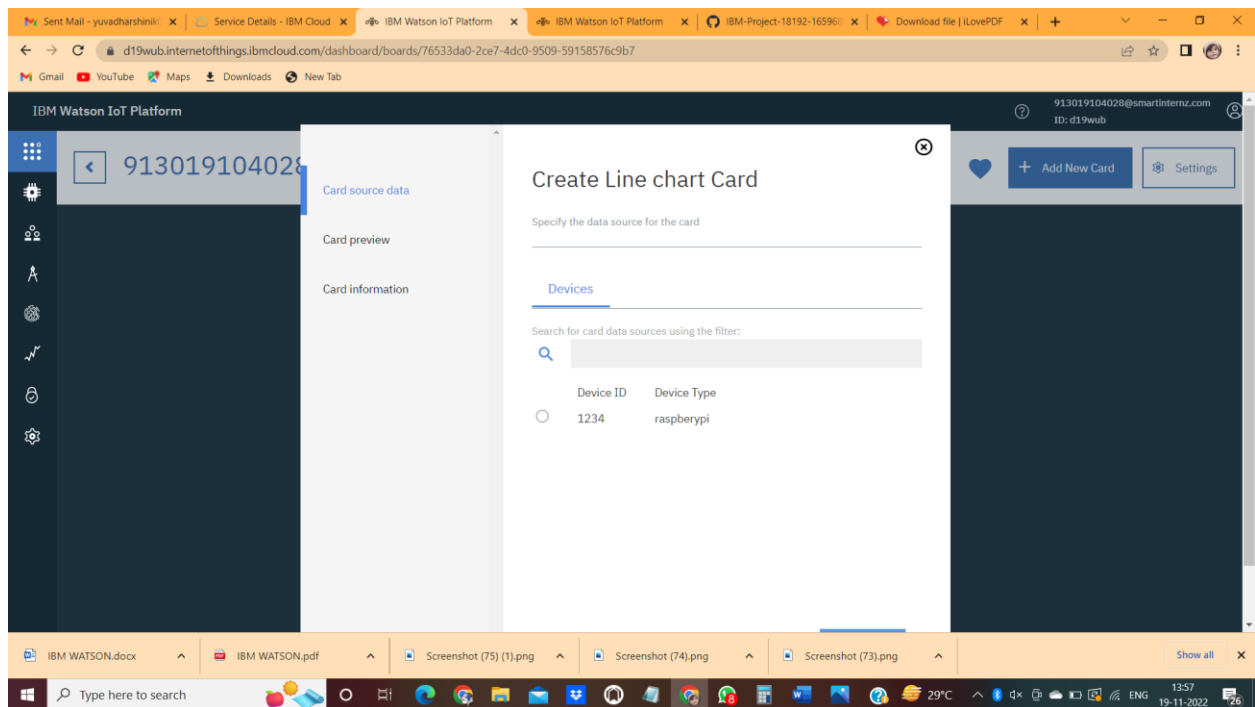
➤ Click on add new card.



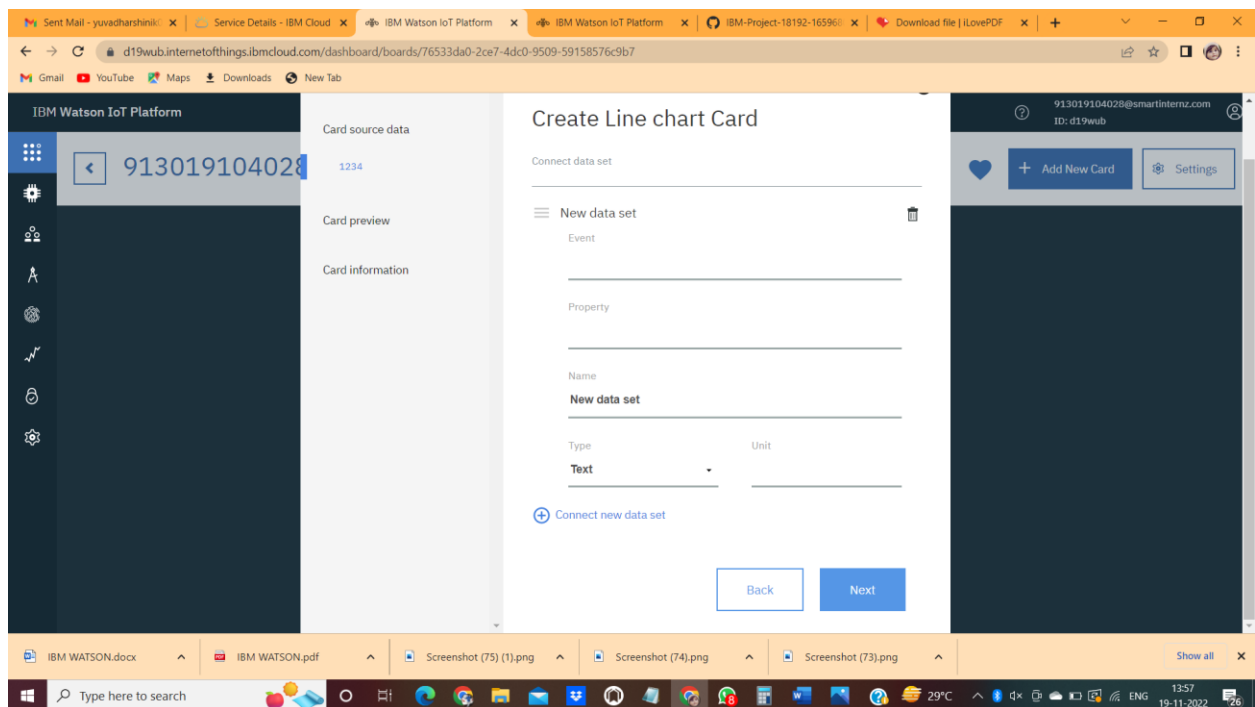
➤ Select the type of Graph u want accordingly and click next



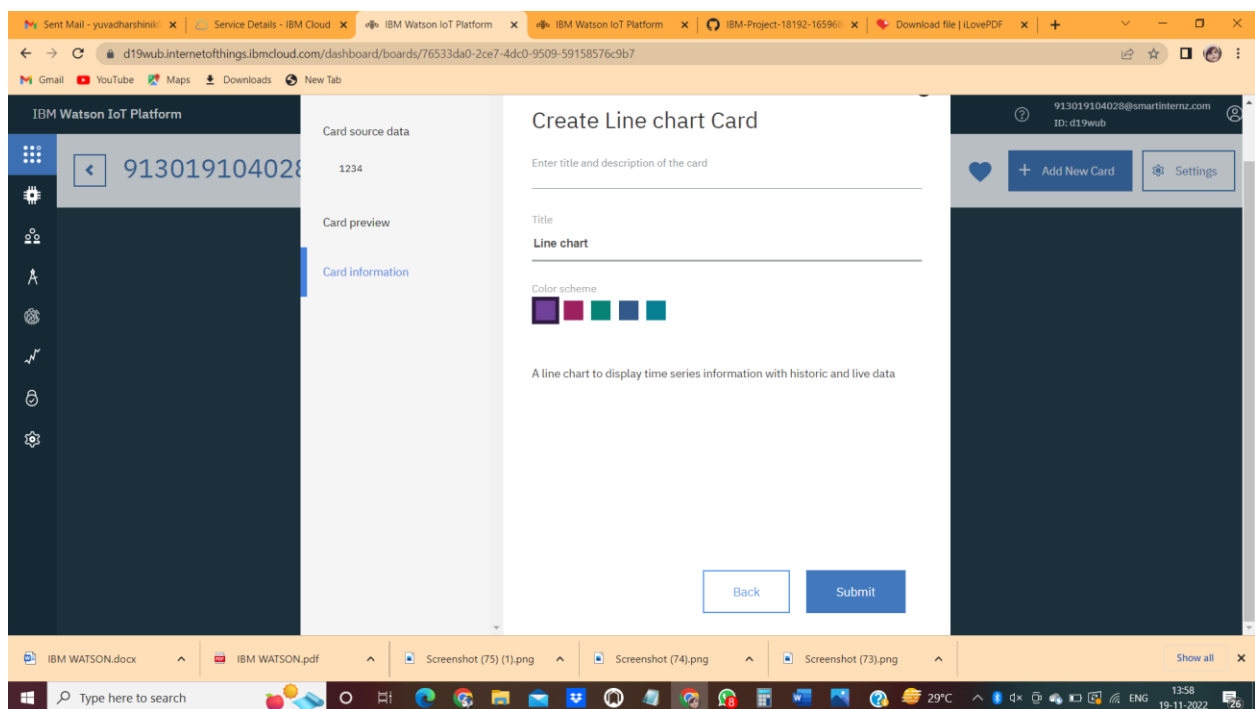
➤ You get the below window, choose the Device and click on Next.



➤ Select the event, properly to be visualized on your graph and click next. In my case it is humidity.

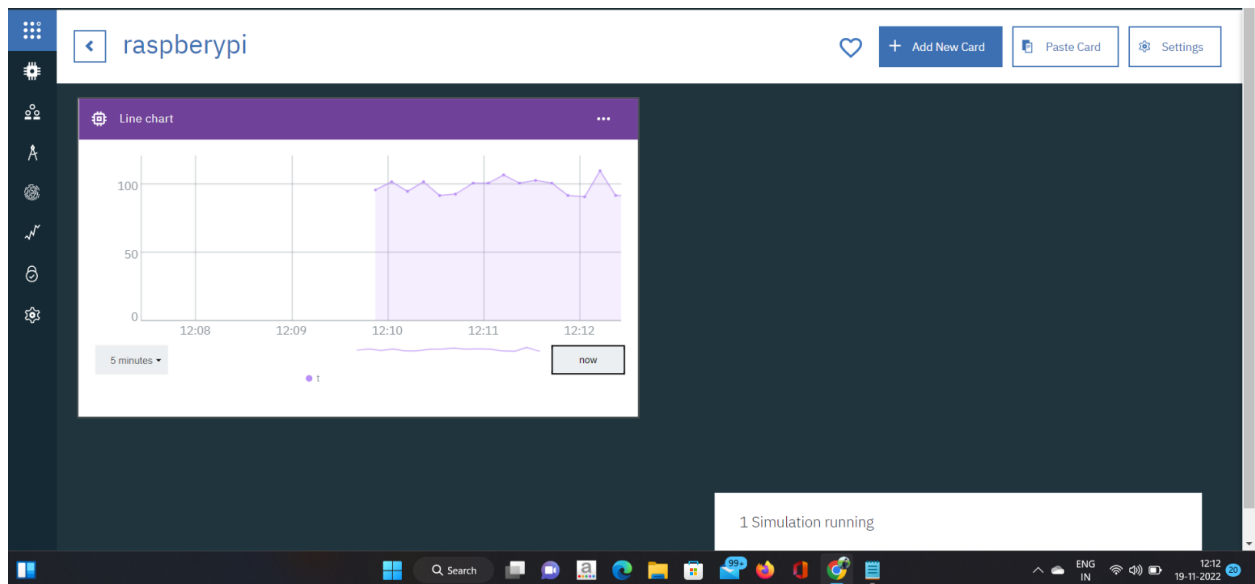


➤ Then select the size of the graph and color of the graph board you want and click next.





➤ Here is the graph



## **RESULT:**

Hence, we were able to send data from our pi to IBM Watson and visualize it on a graph.