

# SENDING DATA FROM RASPBERRY-PI TO IBM WATSON

Team ID	PNT2022TMID14128
Project Name	Smart Waste Management Using for Metropolitan cities

## 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

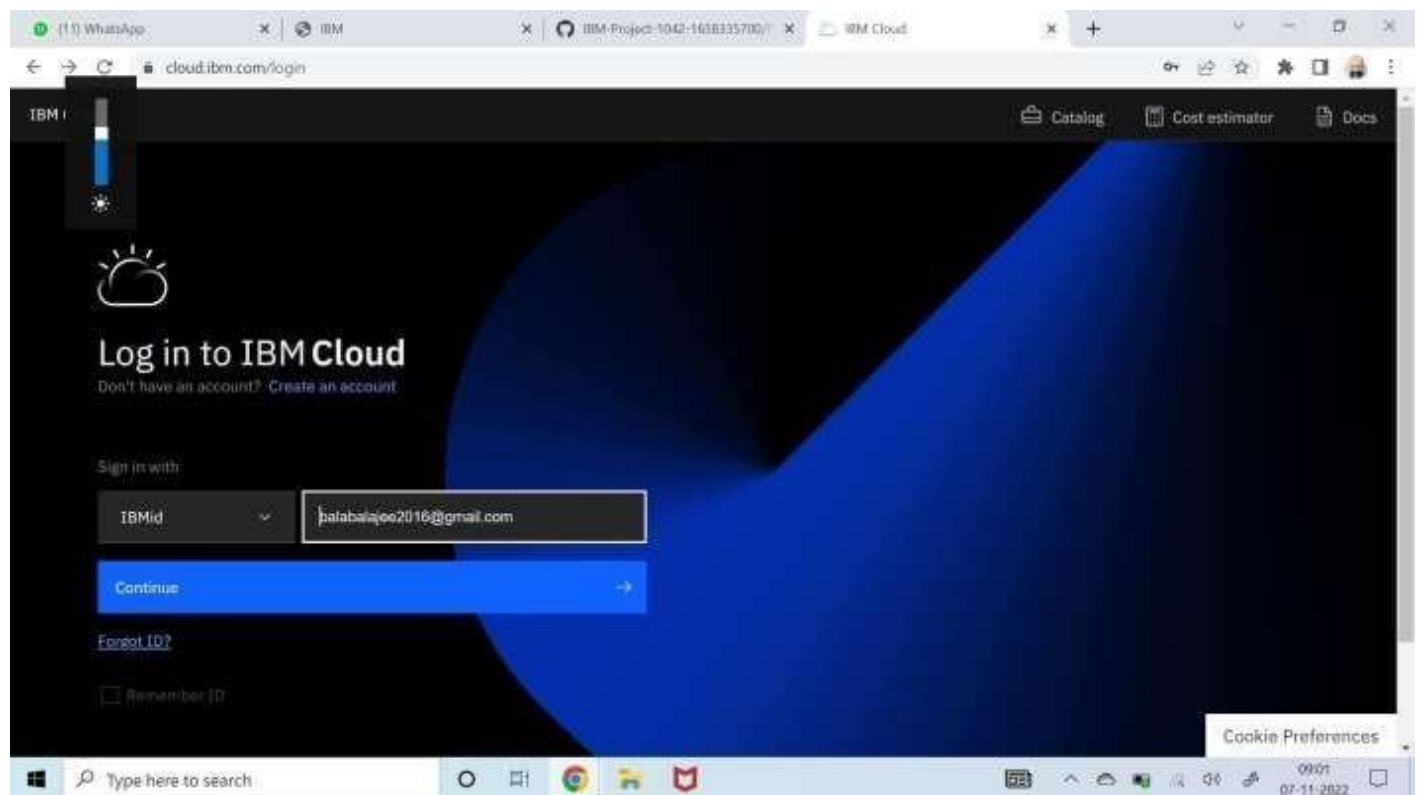
### SOFTWARE:

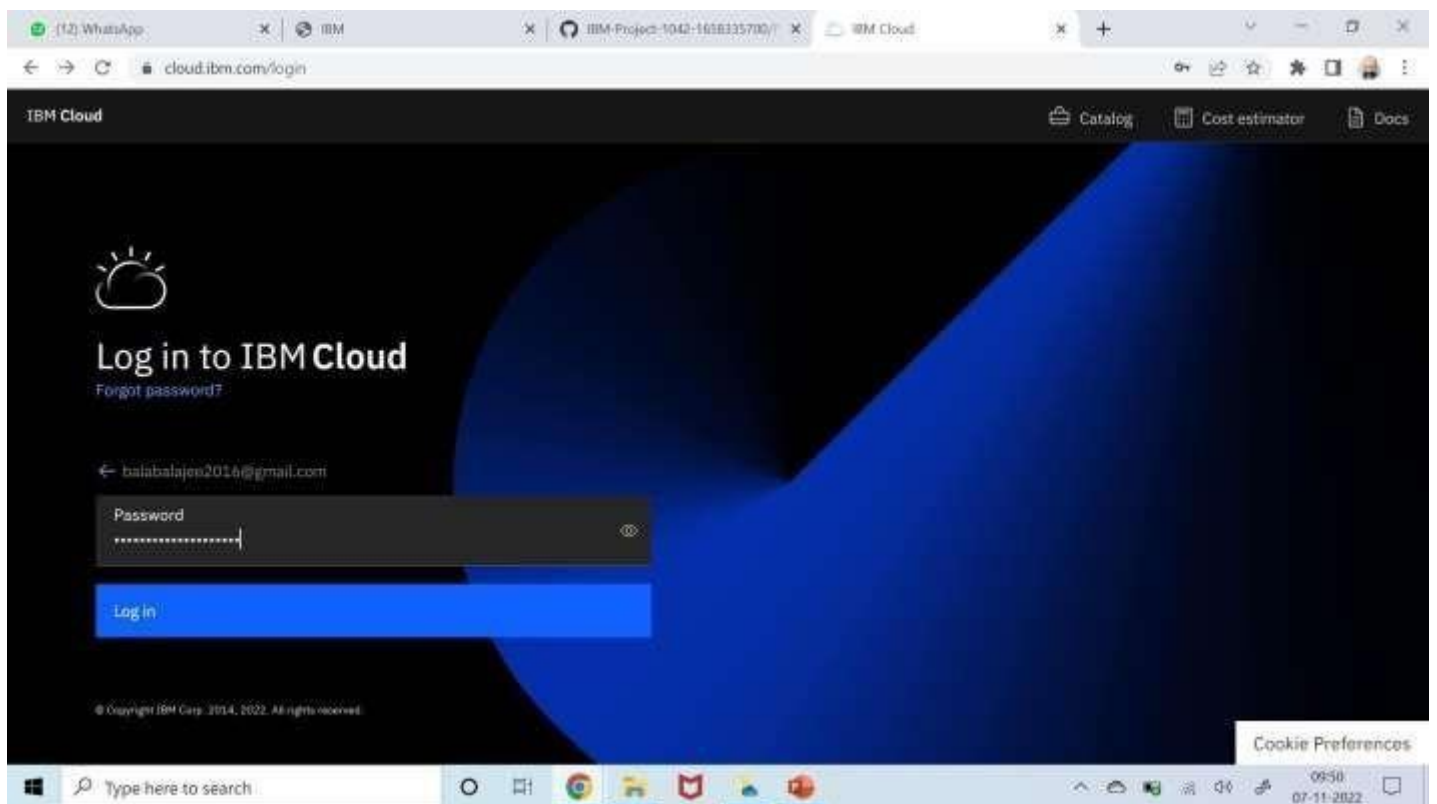
- IBM BLUEMIX ACCOUNT

## STEPS TO BE FOLLOWED

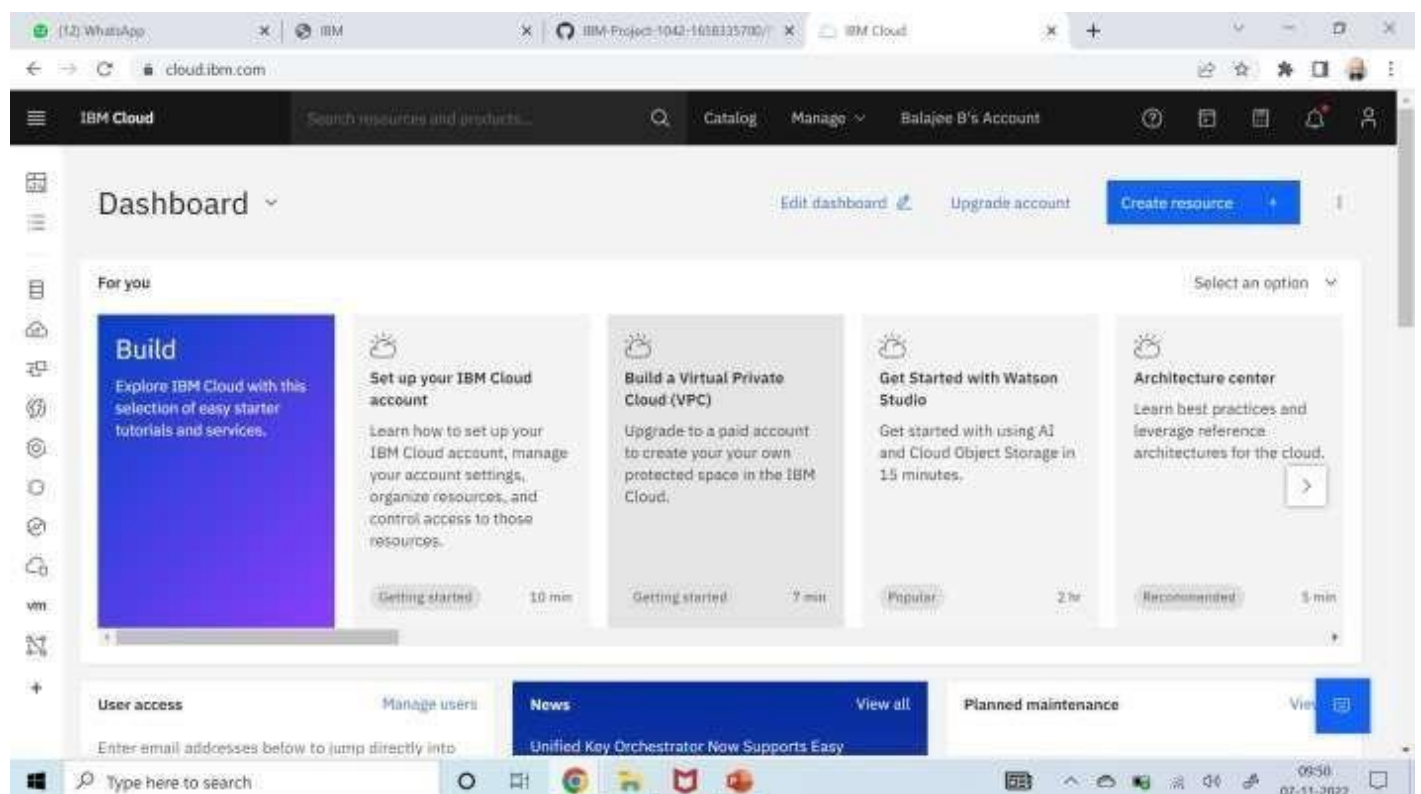
### Step-1: Create a device in IBM Watson:

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

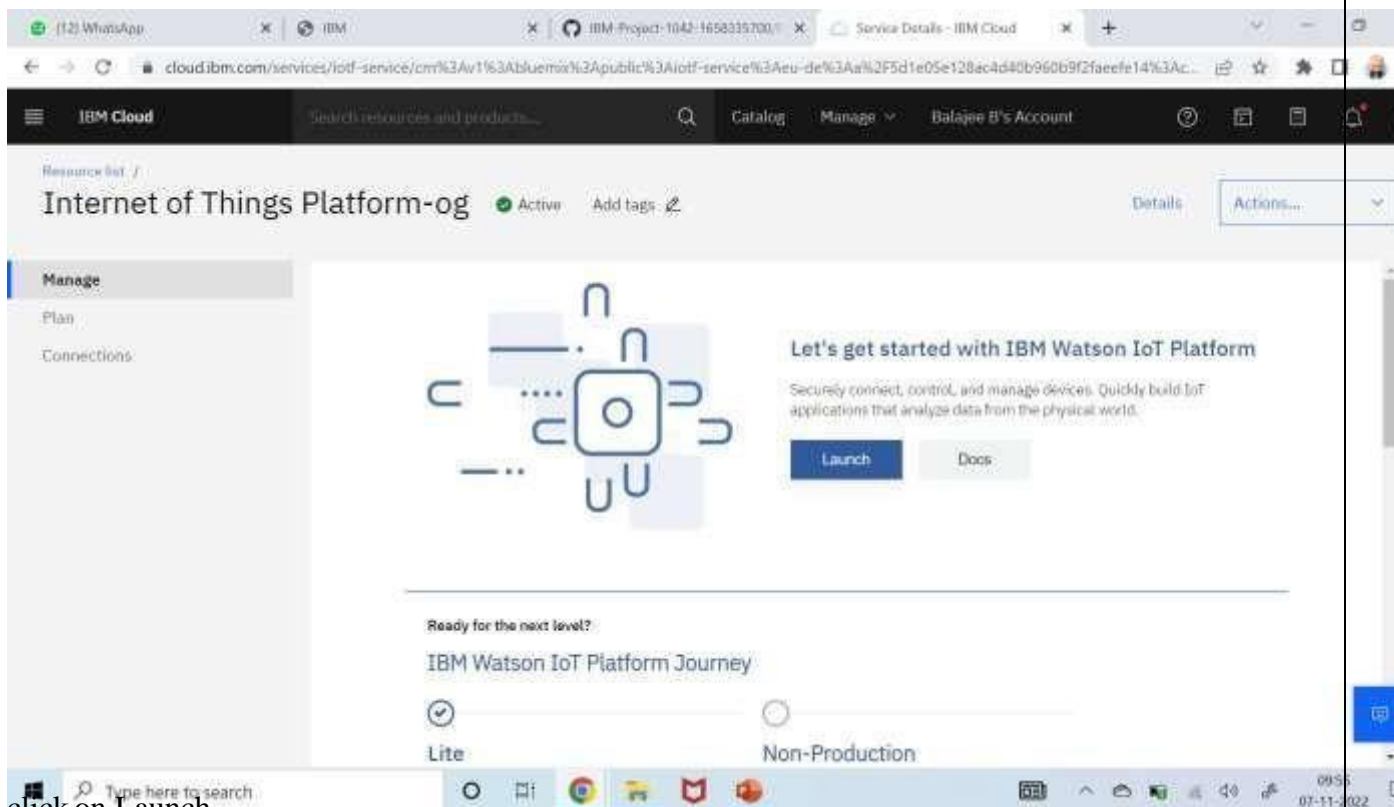
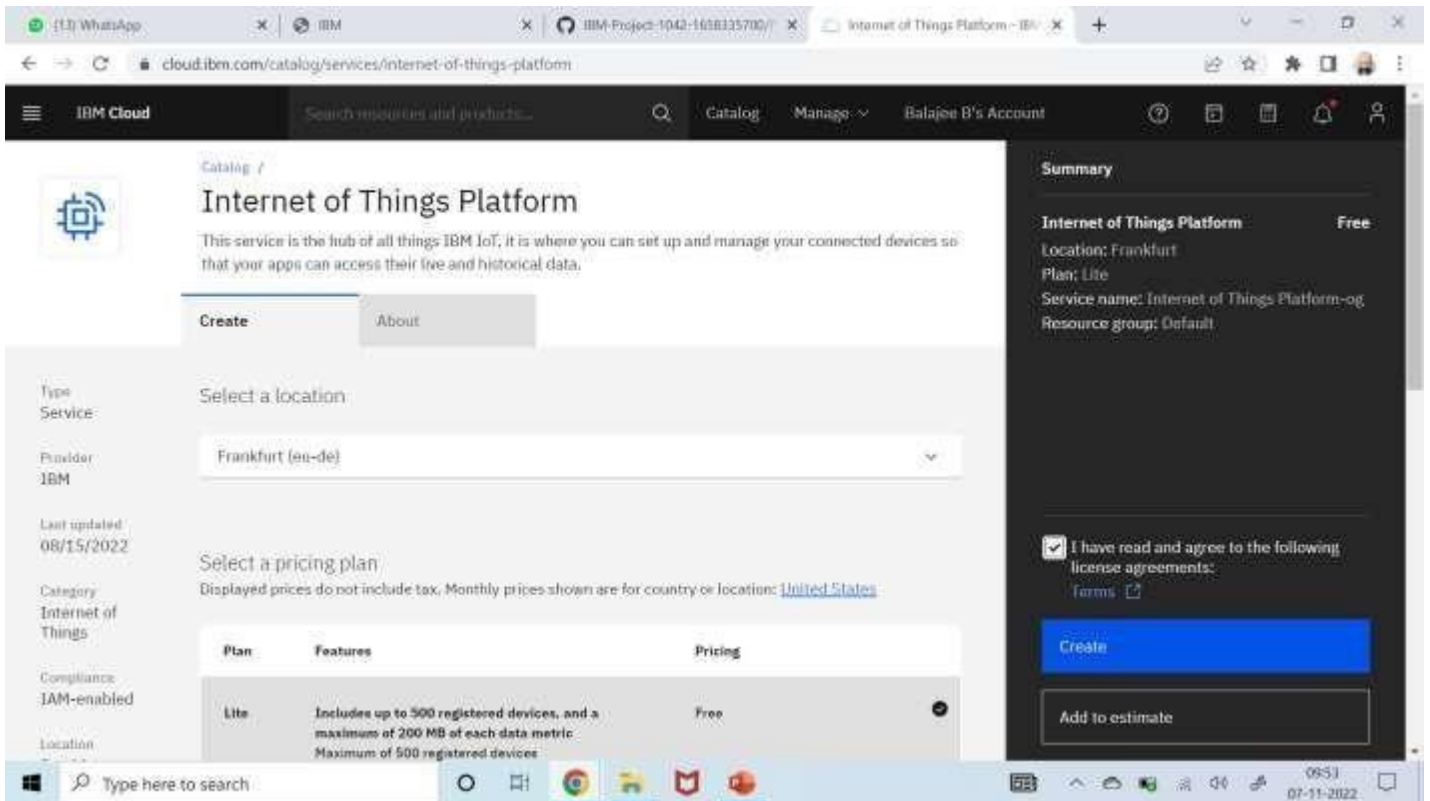




- Click on catalog on your dashboard screen, then under platform go IoT.

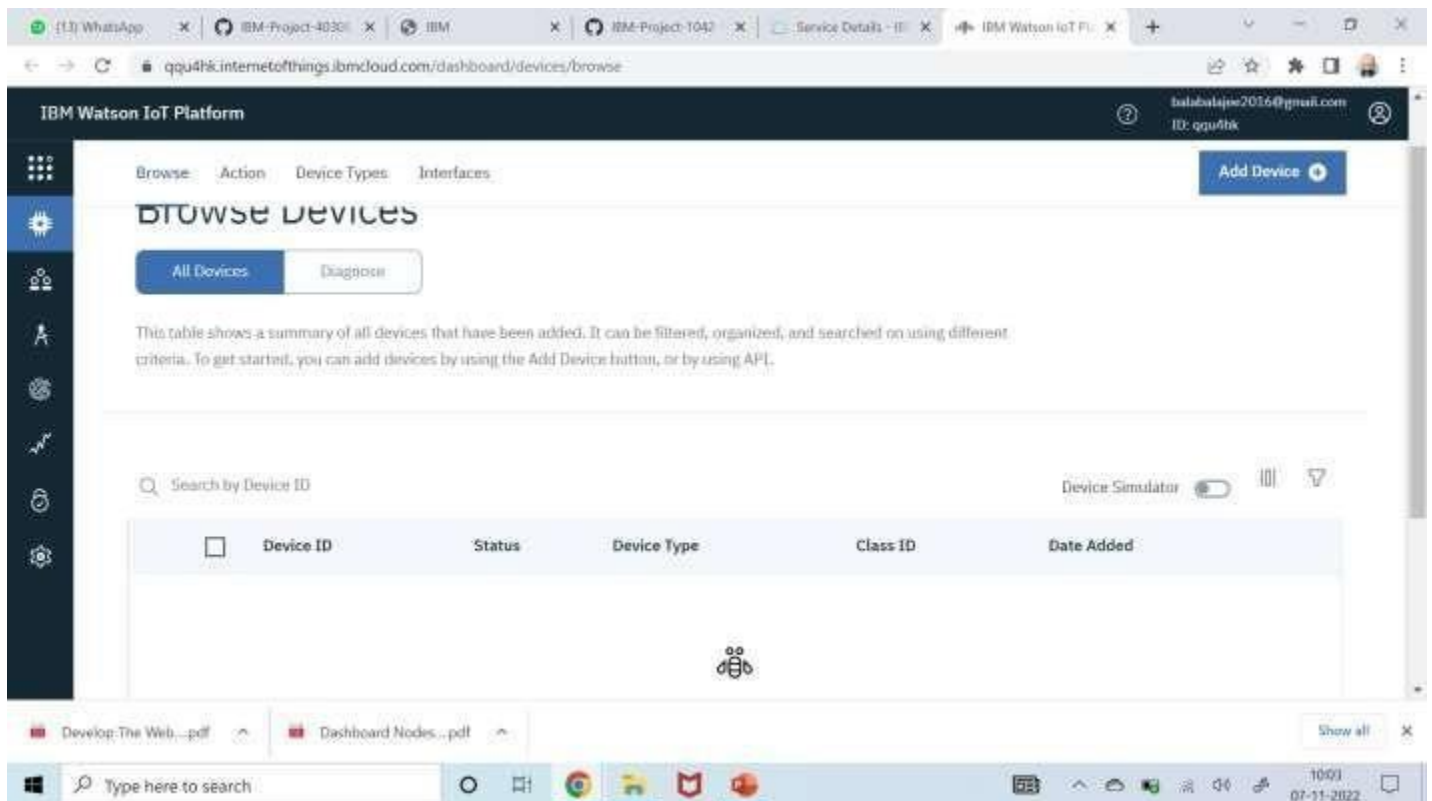


- Check all details and click on create.

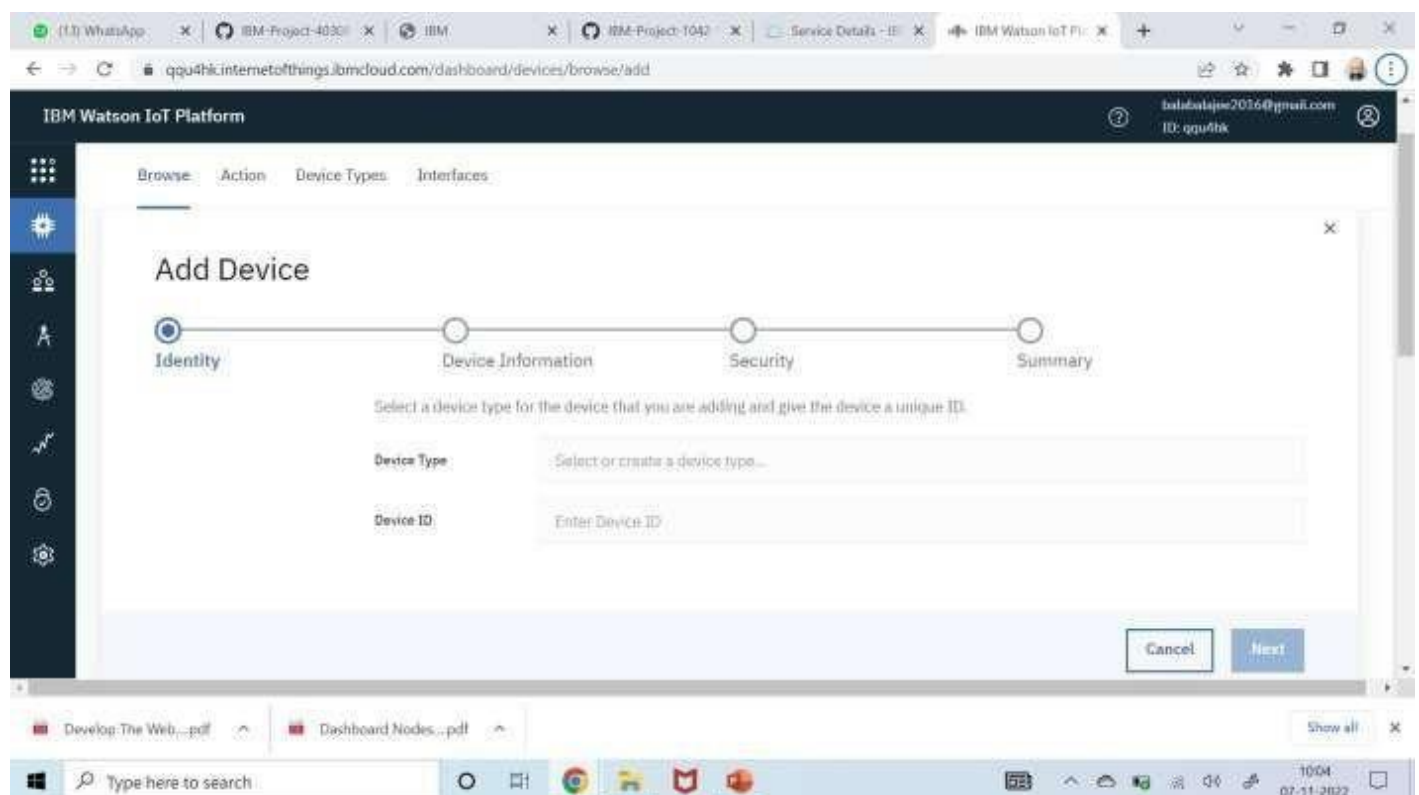


- click on Launch

- 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.

IBM Watson IoT Platform

qqu4hk.internetofthings.ibmcloud.com/dashboard/devices/types/add

balabalajee2016@gmail.com  
ID: qqu4hk

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: ☒ Device Or ☐ Gateway

Name: Fantastic-4

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:

Develop: The Web...pdf Dashboard Nodes...pdf Show all

Type here to search

10:13 07-11-2022

- Click on Finish

IBM Watson IoT Platform

qqu4hk.internetofthings.ibmcloud.com/dashboard/devices/types/add

balabalajee2016@gmail.com  
ID: qqu4hk

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: ☒ Device Or ☐ Gateway

Name: Fantastic-4

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:

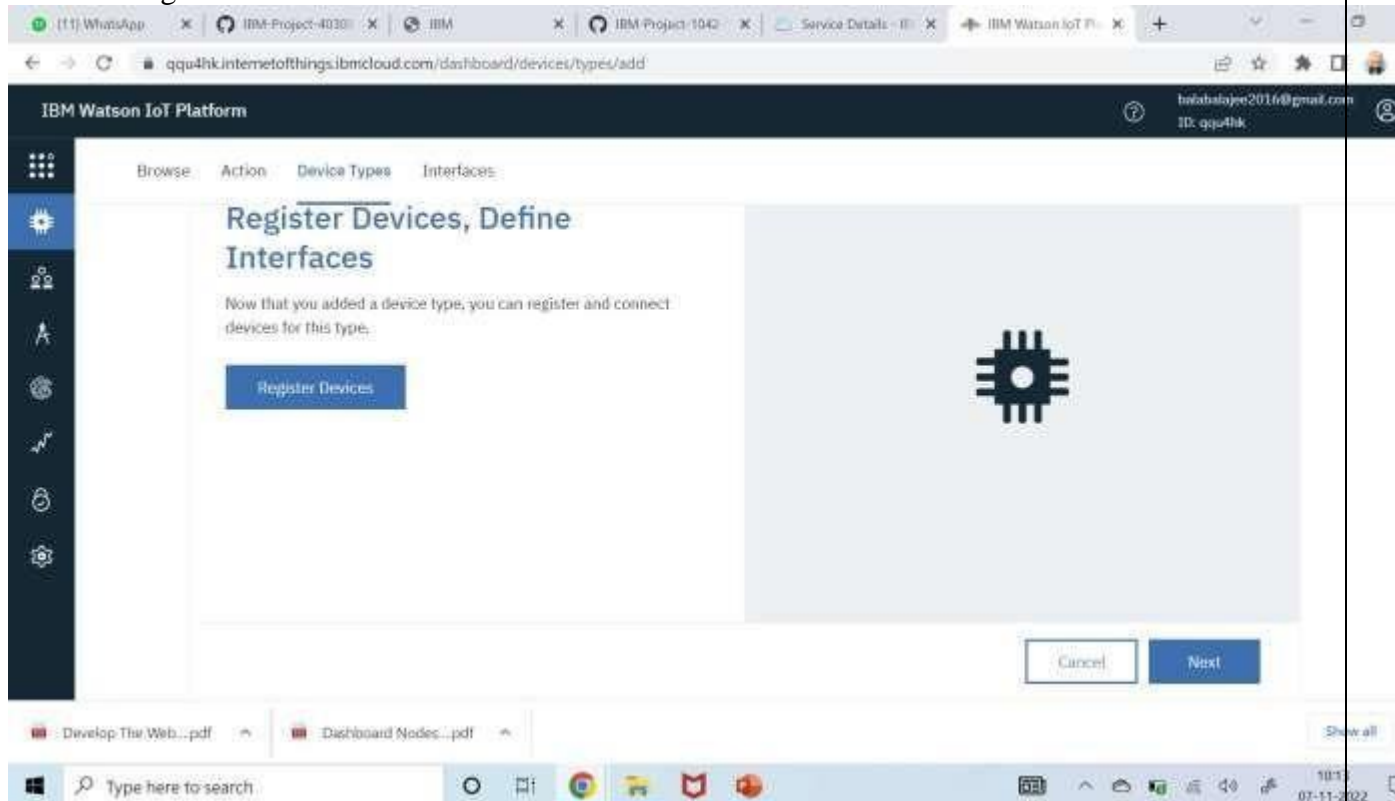
Develop: The Web...pdf Dashboard Nodes...pdf Show all

Type here to search

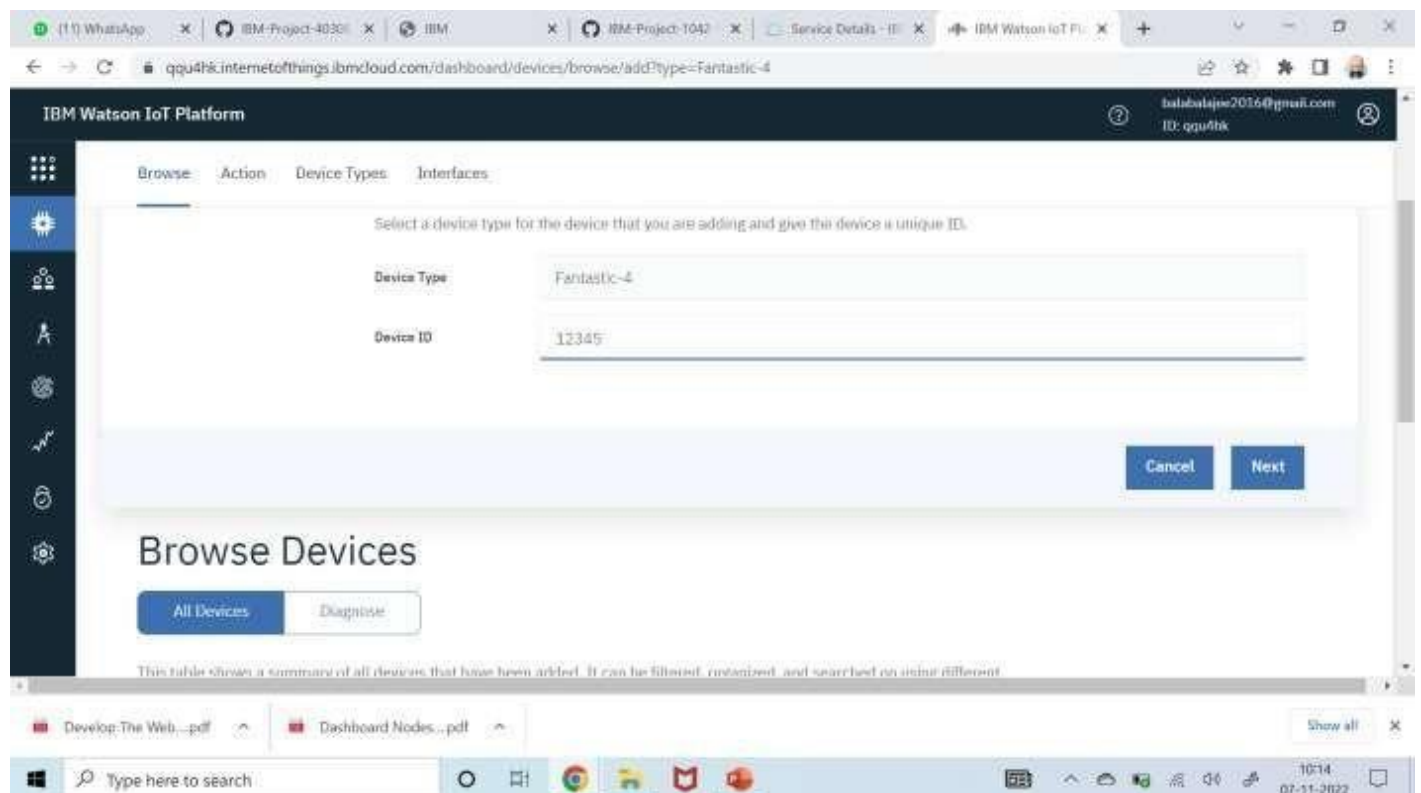
10:13 07-11-2022



- Click on Register Device.



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



➤ Click on Next

The screenshot shows the 'Add Metadata' form in the IBM Watson IoT Platform. The form is titled 'You can modify the default device information and enter more information about the device for identification purposes.' It contains two columns of input fields. The left column includes 'Serial Number', 'Model', 'Description', and 'Hardware Version'. The right column includes 'Manufacturer', 'Device Class', 'Firmware Version', and 'Descriptive Location'. Each field has a placeholder text 'Enter [Field Name]'. Below the left column is an 'Add Metadata' button with a plus icon. At the bottom right are 'Back' and 'Next' buttons. The browser's address bar shows the URL 'qqu4hk.internetofthings.ibmcloud.com/dashboard/devices/browse/add?type=Fantastic-4'. The user's email 'balabala2016@gmail.com' and ID 'qqu4hk' are visible in the top right corner.

IBM Watson IoT Platform

balabala2016@gmail.com  
ID: qqu4hk

Browse Action Device Types Interfaces

You can modify the default device information and enter more information about the device for identification purposes.

Serial Number Enter Serial Number

Model Enter Model

Description Enter Description

Hardware Version Enter Hardware Version

Manufacturer Enter Manufacturer

Device Class Enter Device Class

Firmware Version Enter Firmware Version

Descriptive Location Enter Descriptive Location

Add Metadata +

Back Next

Develop: The Web...pdf Dashboard Nodes...pdf Show all

Type here to search

10:14  
07-11-2022

➤ Click on Next

The screenshot shows the 'Authentication Token' selection screen in the IBM Watson IoT Platform. It presents two options: 'Auto-generated authentication token (default)' and 'Self-provided authentication token'. The 'Auto-generated' option explains that the service will generate an 18-character token. The 'Self-provided' option requires a user-defined token between 8 and 36 characters. Below these options is an 'Authentication Token' input field with a placeholder 'Enter an optional token'. A note states: 'Make a note of the generated token. Lost authentication tokens cannot be recovered. Tokens are encrypted before being stored.' Another note at the bottom says: 'Authentication tokens are encrypted before we store them.' The browser's address bar shows the URL 'qqu4hk.internetofthings.ibmcloud.com/dashboard/devices/browse/add?type=Fantastic-4'. The user's email 'balabala2016@gmail.com' and ID 'qqu4hk' are visible in the top right corner.

IBM Watson IoT Platform

balabala2016@gmail.com  
ID: qqu4hk

Browse Action Device Types Interfaces

There are two options for selecting a device authentication token.

**Auto-generated authentication token (default)**

Allow the service to generate an authentication token for you. Tokens are 18 characters and contain a mix of alphanumeric characters and symbols. The token is returned to you at the end of the device registration process.

**Self-provided authentication token**

Provide your own authentication token for this device. The token must be between 8 and 36 characters and contain a mix of lowercase and uppercase letters, numbers, and symbols, which can include hyphens, underscores, and periods. Do not use repeated characters, dictionary words, user names, or other predefined sequences.

Authentication Token Enter an optional token

Make a note of the generated token. Lost authentication tokens cannot be recovered. Tokens are encrypted before being stored.

Authentication tokens are encrypted before we store them.

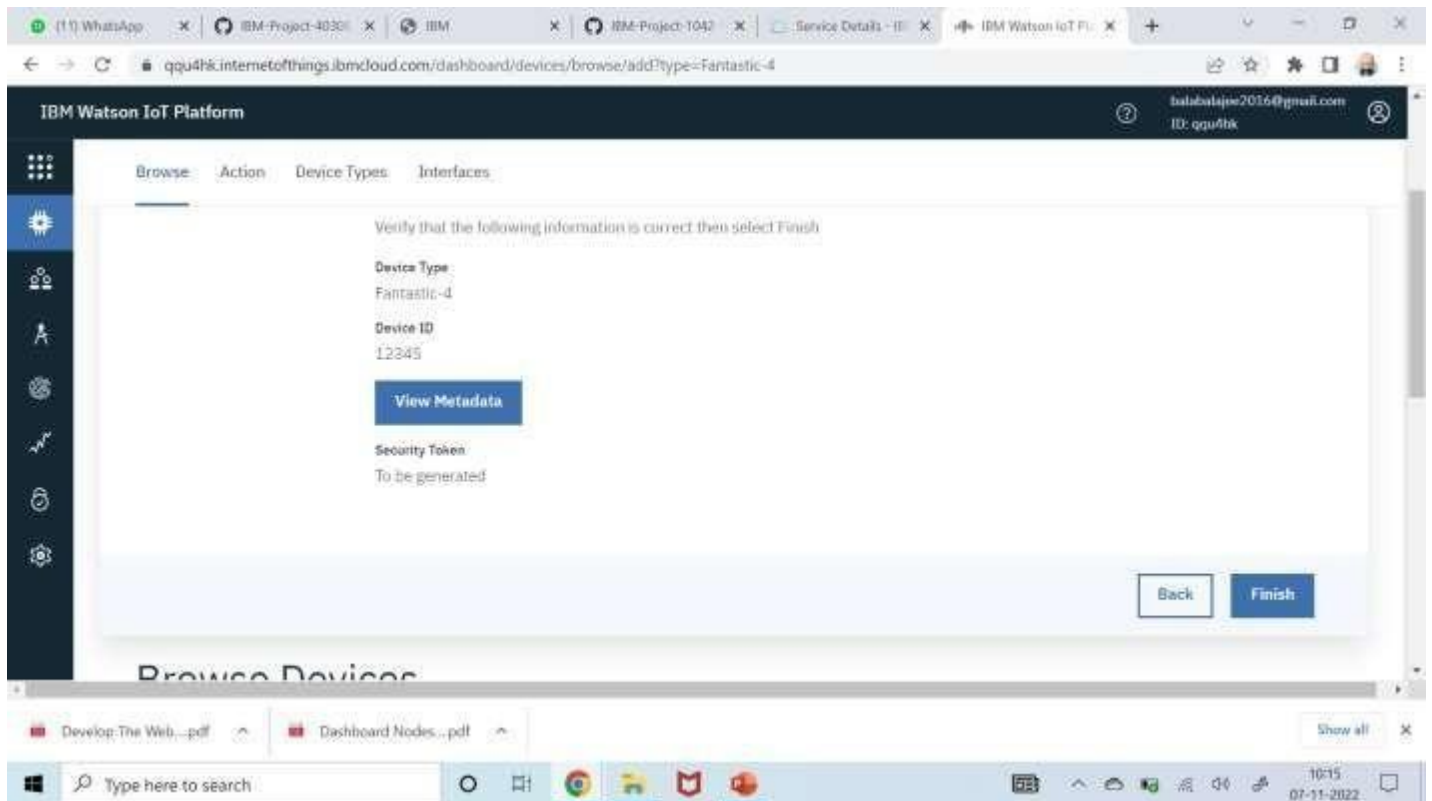
Develop: The Web...pdf Dashboard Nodes...pdf Show all

Type here to search

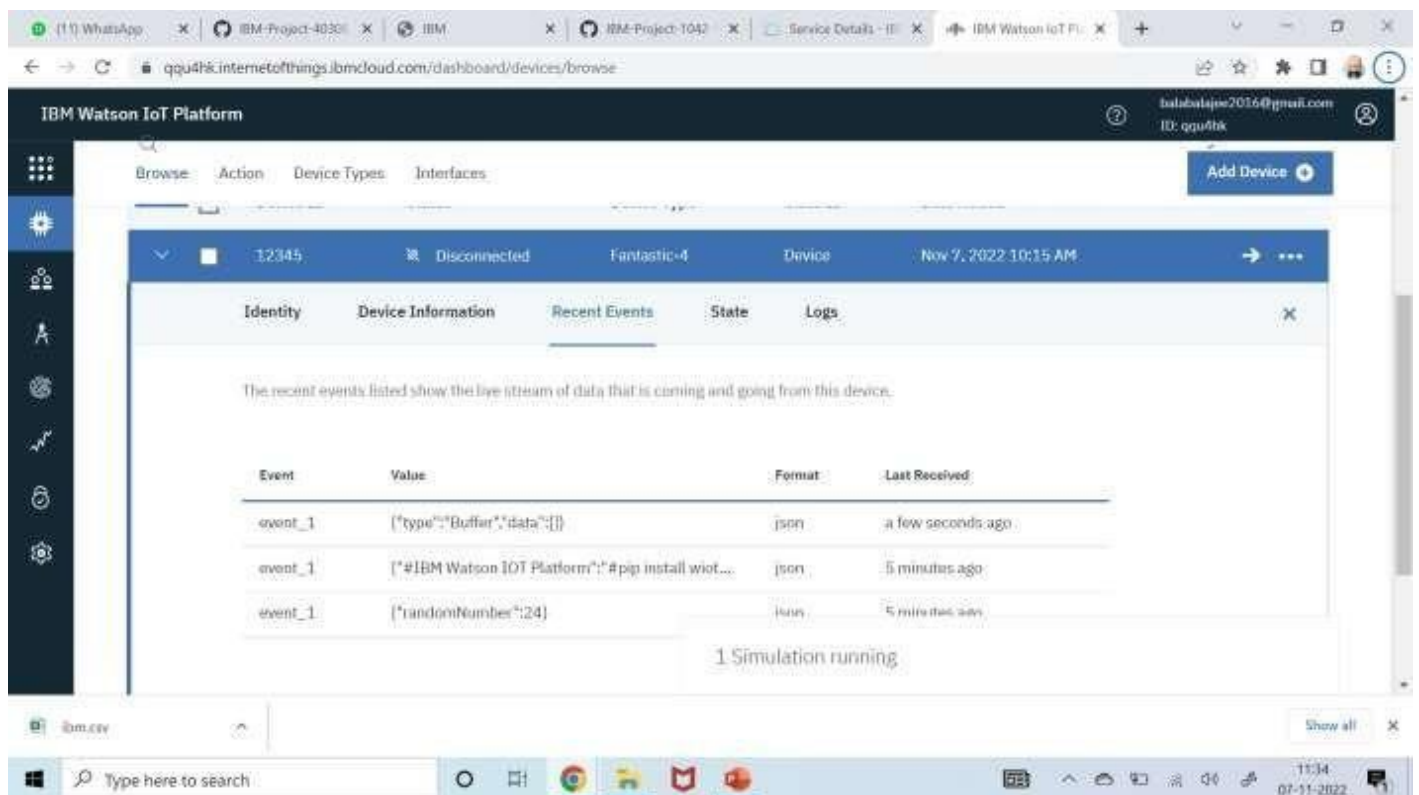
10:14  
07-11-2022



➤ Click on Finish



➤ Device is created



## 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`

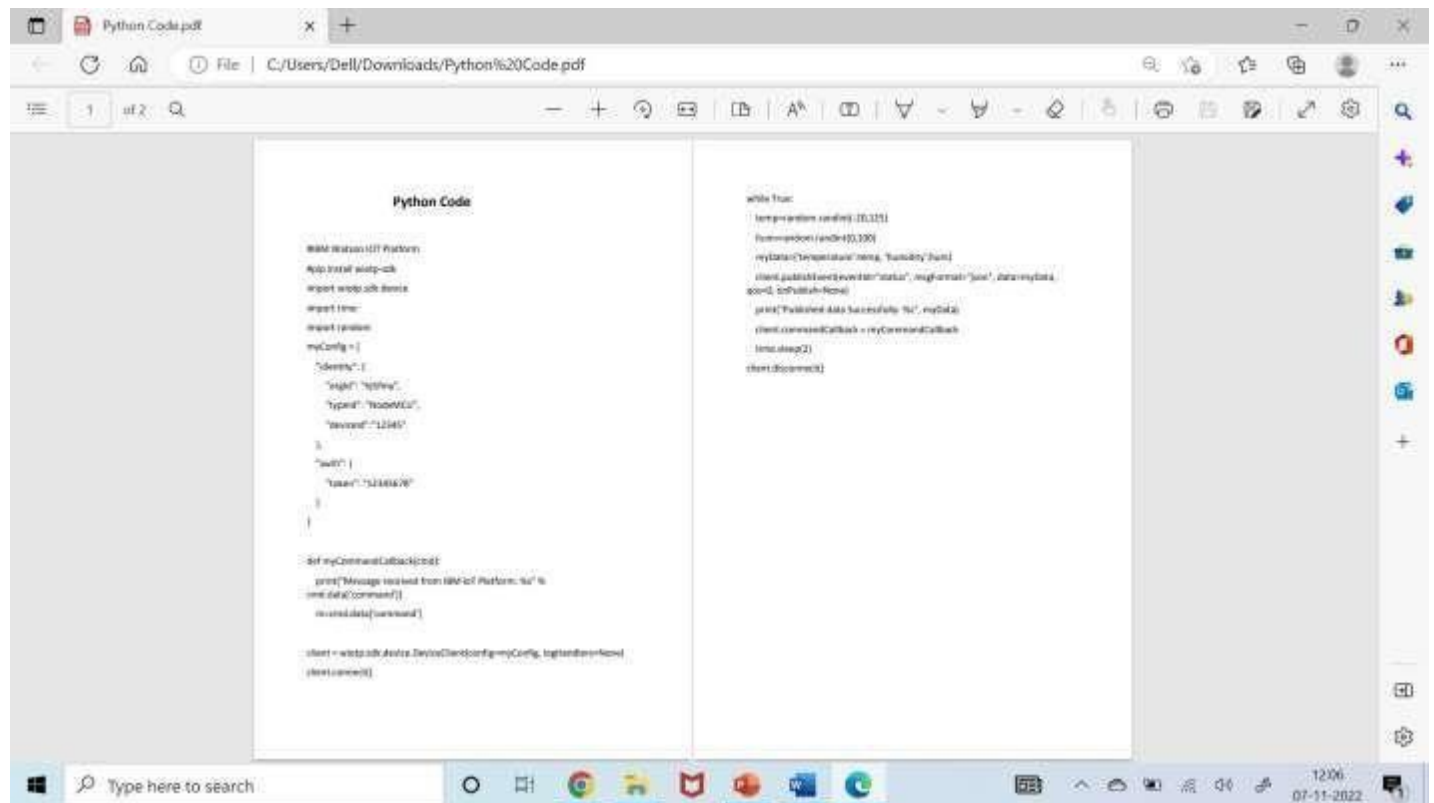
Following are the images as to what appears on your pi's terminal when u type these commands

[illegible]

- Then open your terminal and type `pip install ibmiotf`

[illegible]

- I have sent DHT-11 Sensors data to ibm bluemix .To get the code u need to login into IOT GYAN.
- Then I get the image as follows in my pi's shell:



```
Python Code

#IBM Watson IoT Platform
App install pip3-cb
import websocket
import time
import random
myConfig = {
    "deviceId": "1",
    "type": "DHT11",
    "typeId": "DHT11",
    "version": "1.0.0"
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd)
    if cmd["command"] == "publish":
        myData = {
            "deviceId": "1",
            "type": "DHT11",
            "typeId": "DHT11",
            "version": "1.0.0"
        }
        myData["data"] = {
            "temp": random.randint(10, 30),
            "humidity": random.randint(40, 90)
        }
        myData["status"] = "success"
        myData["data"] = myData
        print("Published Data Successfully: %s" % myData)
        myCommandCallback(myData)
        time.sleep(2)
        myCommandCallback(myData)

client = websocket.WebSocketApp("ws://iotgateway:mqtt://myConfig", myConfig)
client.connect()
```

### 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.

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 functions. The main content area shows a device card for 'Fantastic-4' (ID: 12345), which is 'Disconnected'. Below the card, the 'Recent Events' tab is active, displaying a table of events:

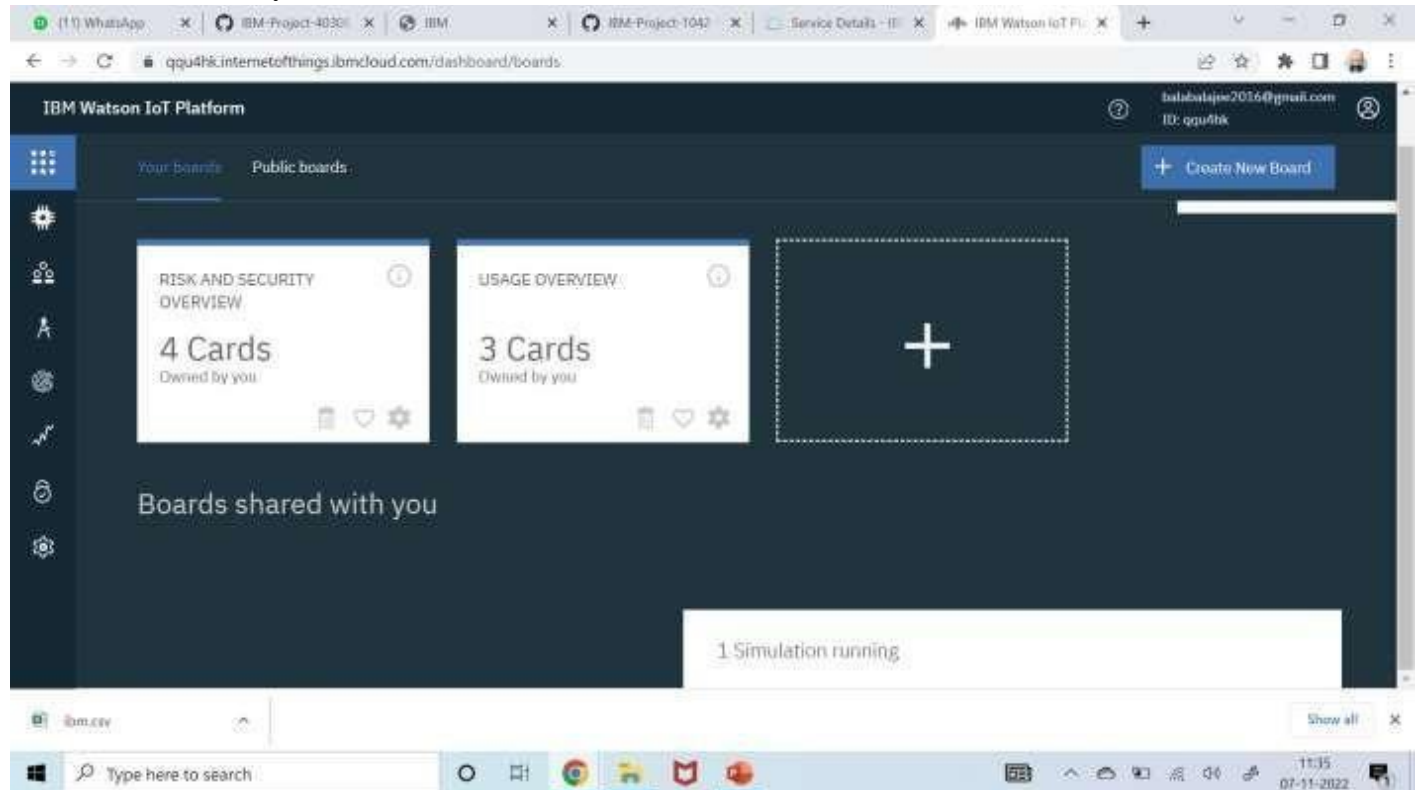
Event	Value	Format	Last Received
event_1	{"type":"Buffer","data":[]}	json	a few seconds ago
event_1	{"#IBM Watson IoT Platform":"#pip install wiot..."}	json	5 minutes ago
event_1	{"randomNumber":24}	json	5 minutes ago

At the bottom of the interface, a status message indicates '1 Simulation running'.

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

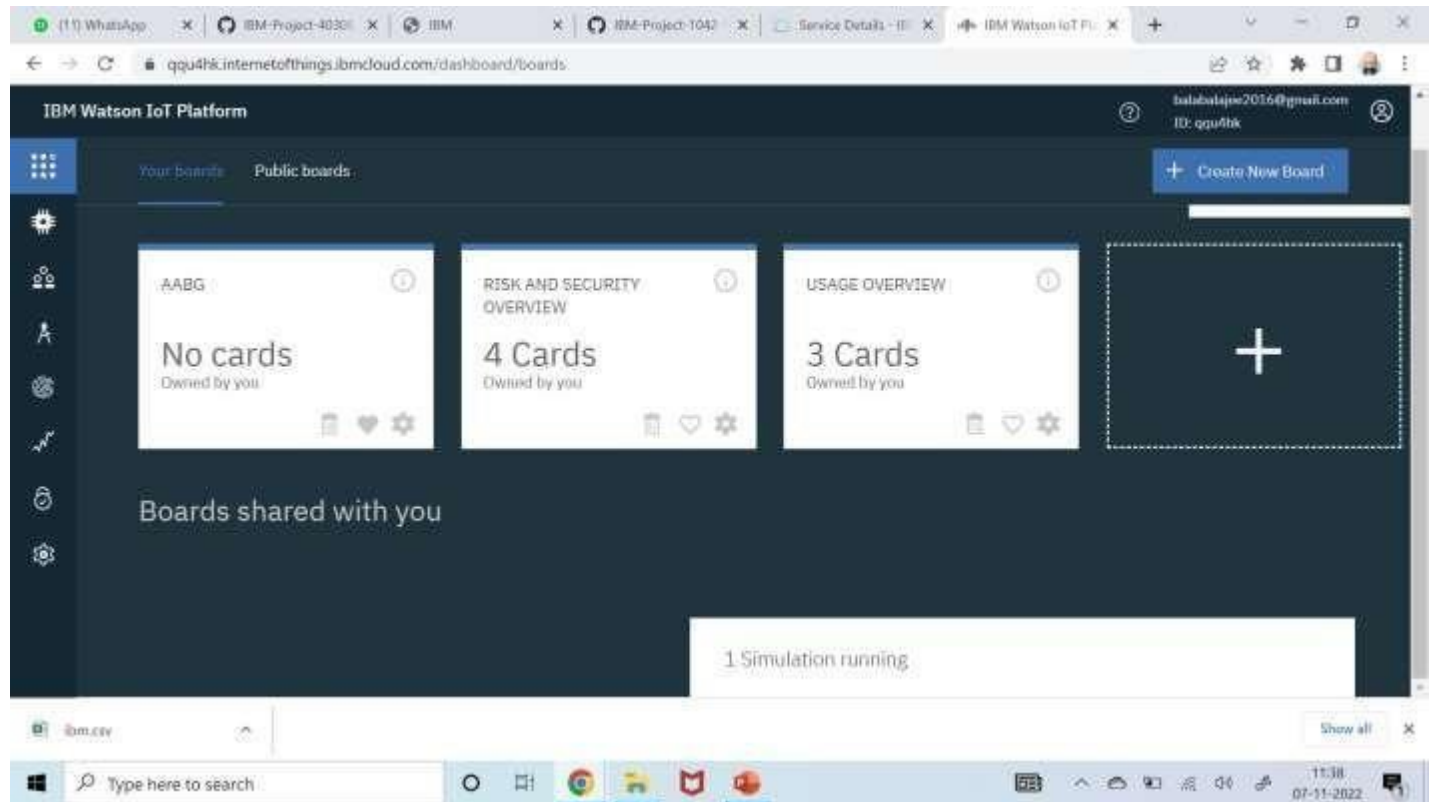
#### Step-4: Creating boards and cards for visualization of data:

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

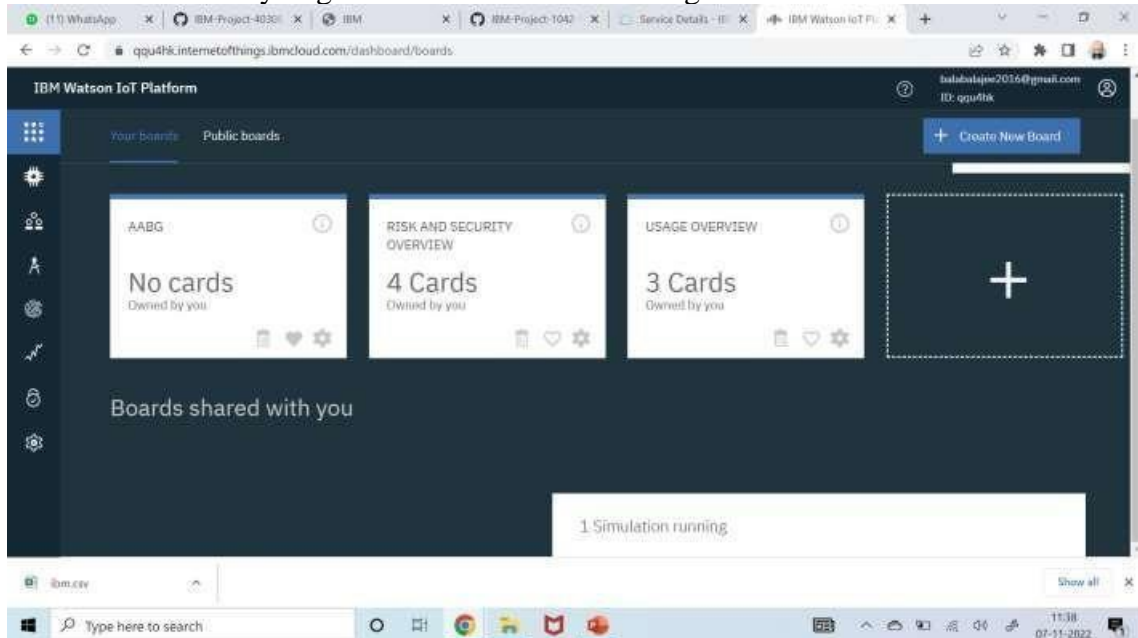


- Click on Create a new board to create a board .

The given below window appears give a name and description to your board as shown in the window below.



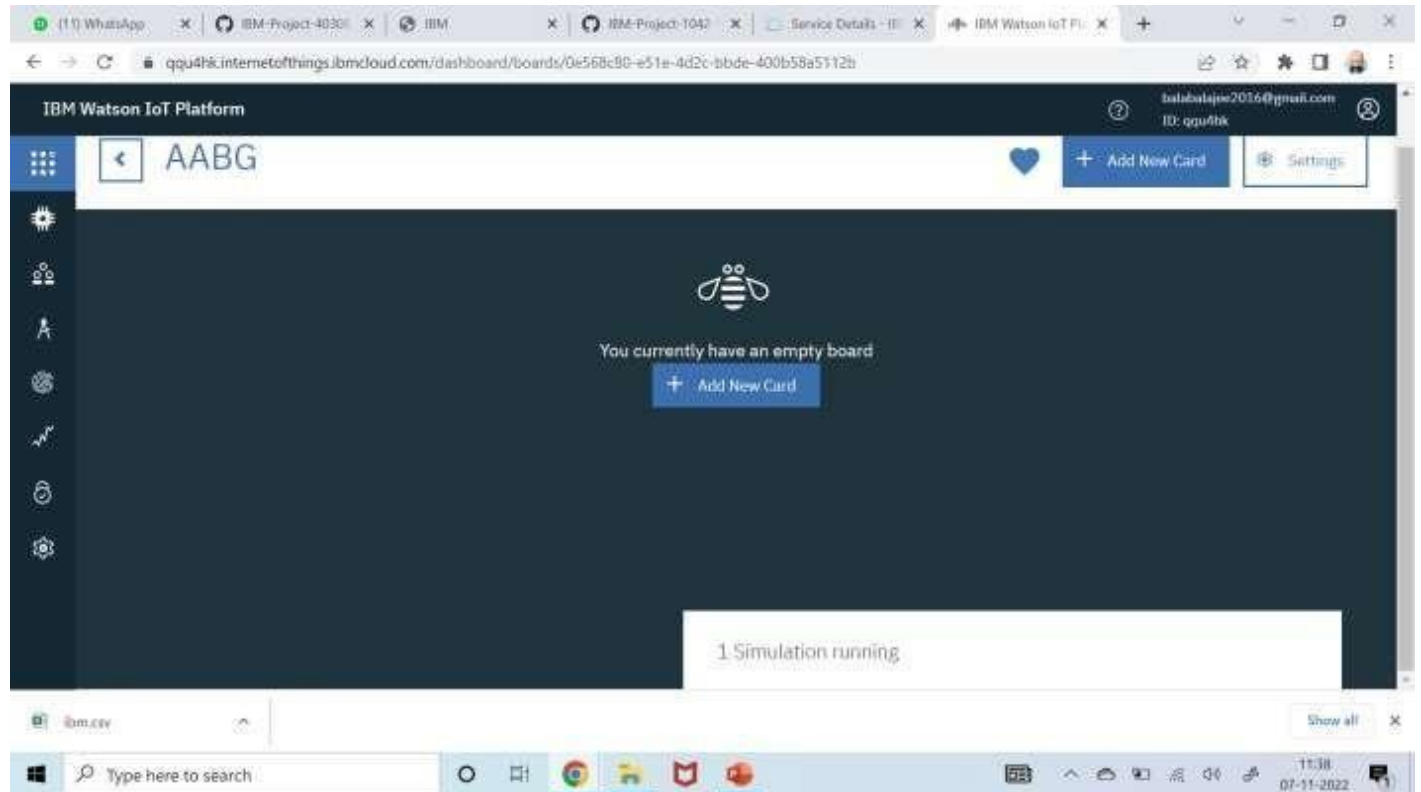
- Then click on Next you get the below window then again click on Sub



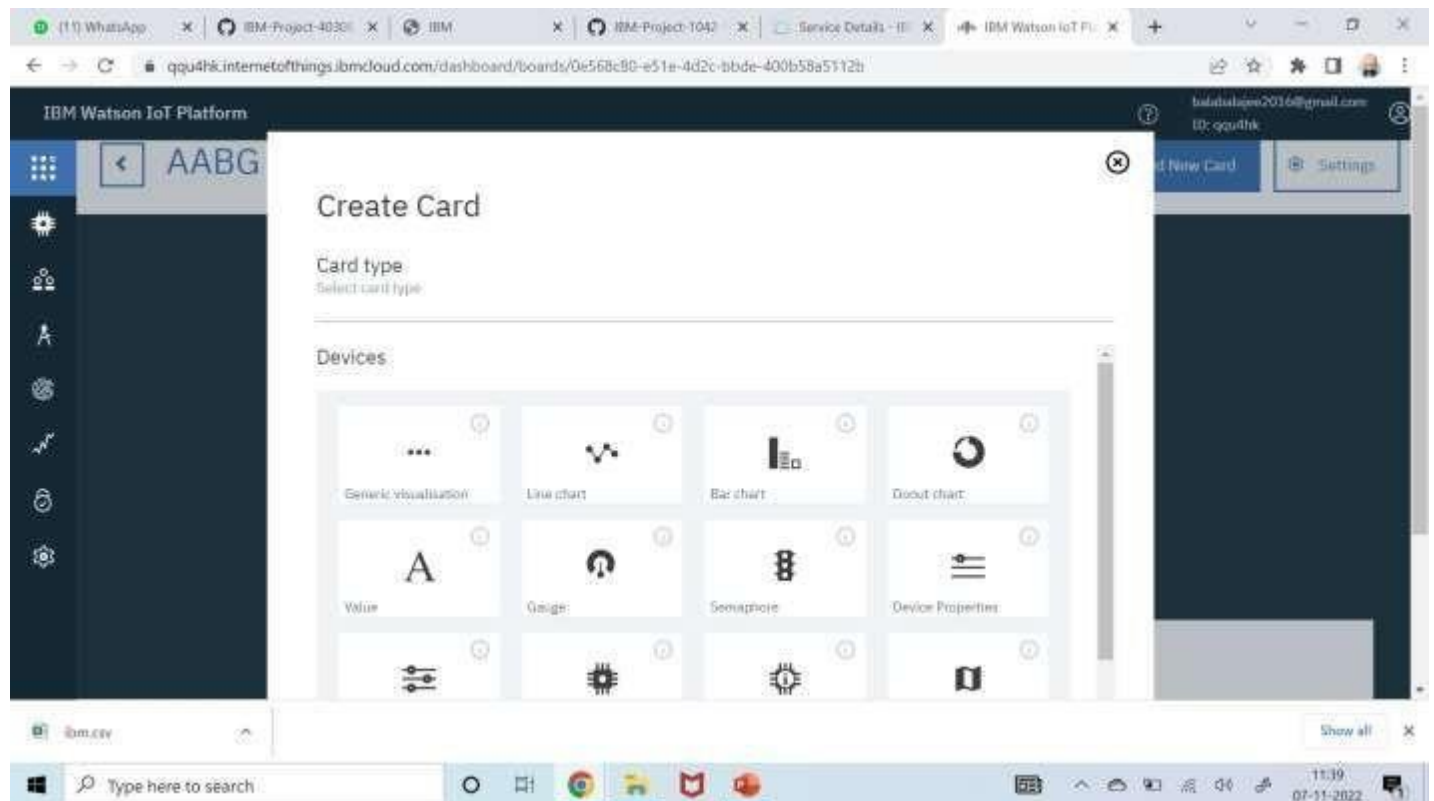


Then double click on your boards name which you have created.

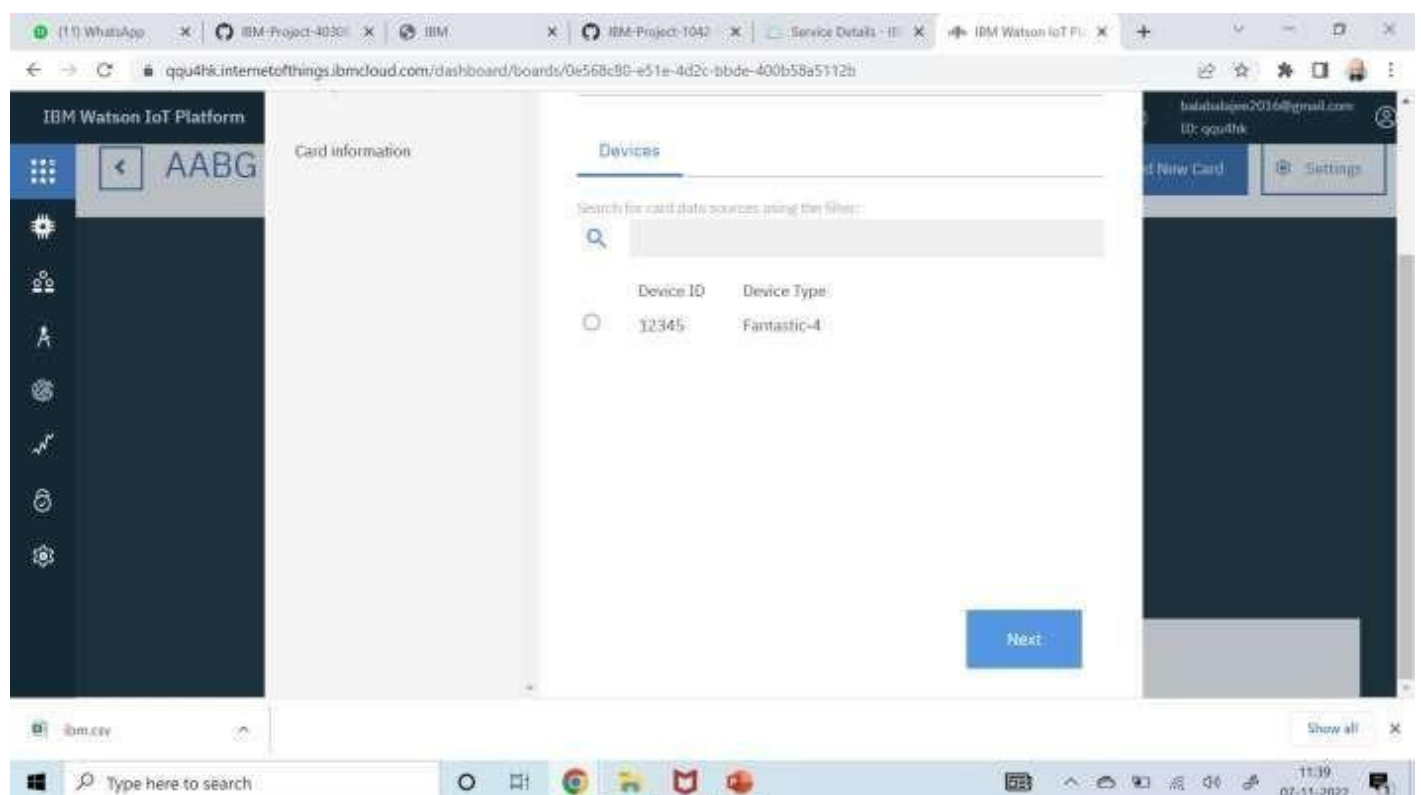
➤ Click on Add New Card



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

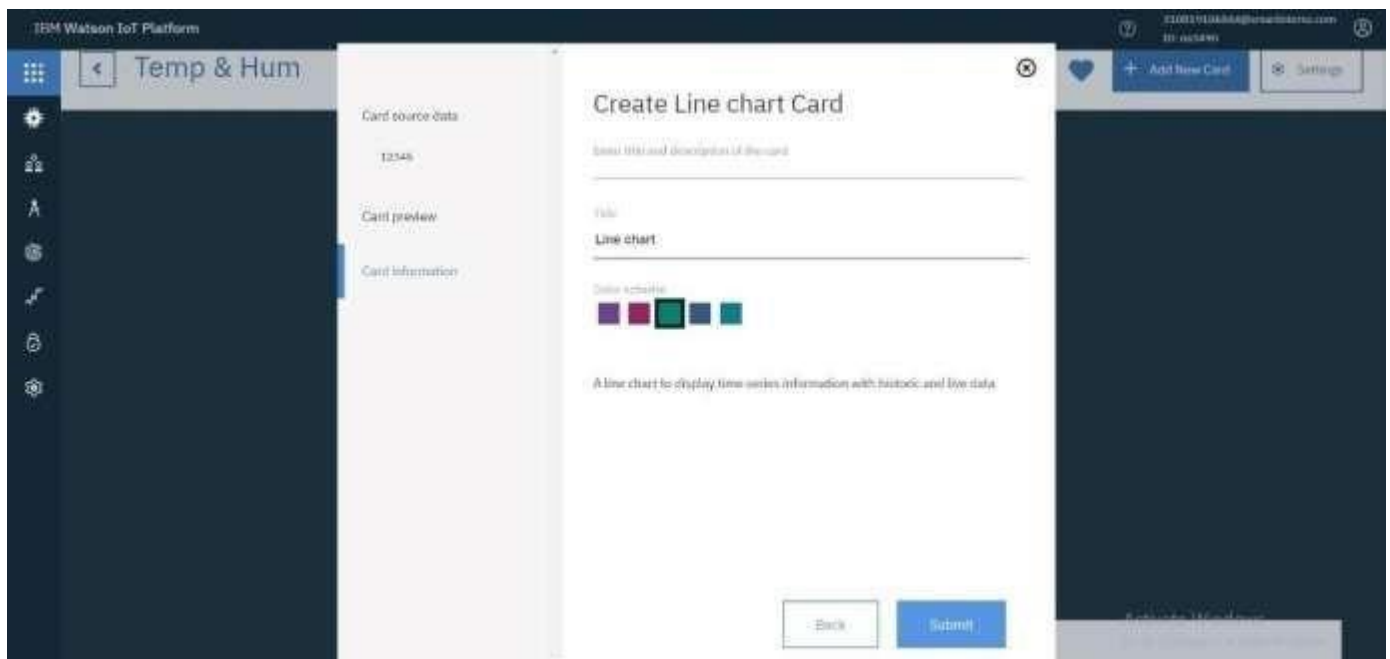


- You get the below window, choose the Device and click on Next.

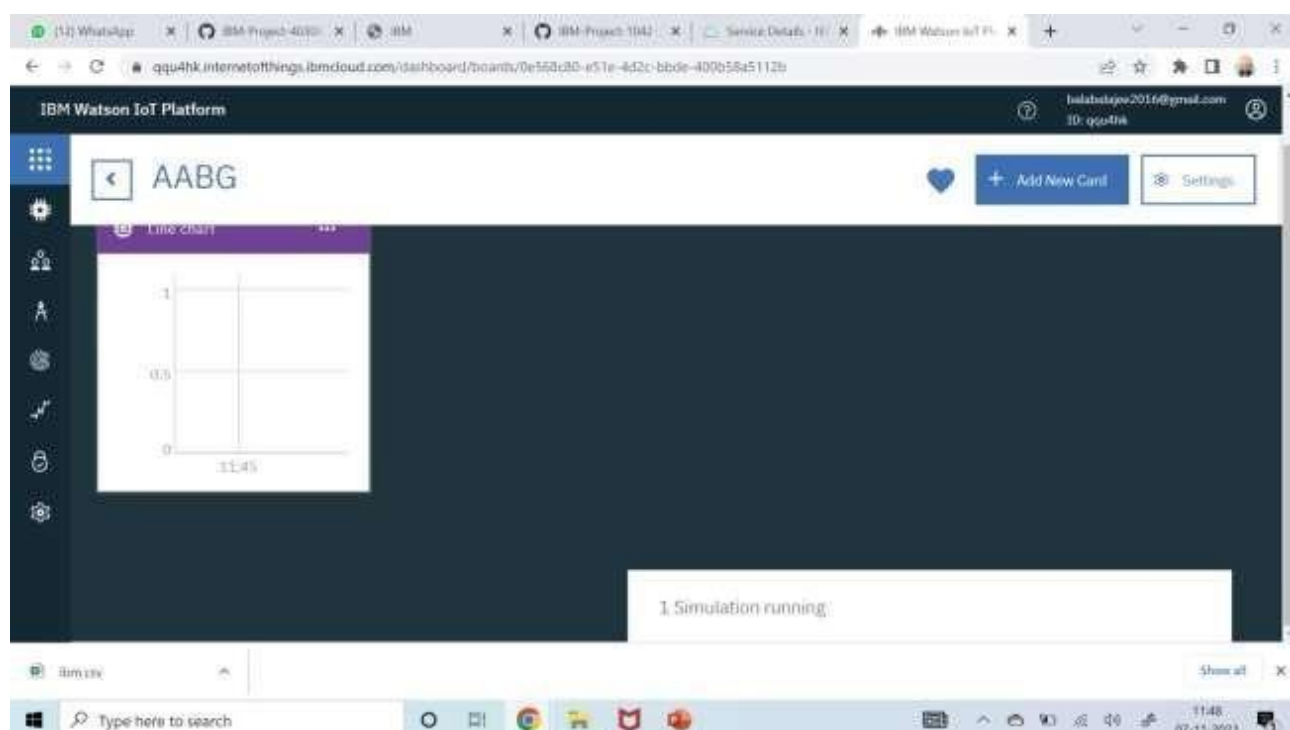


➤ Select the event, properly to be visualized on your graph and click next. In my case it is humidity

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



➤ Here is the graph



- Repeat the process to get different graphs.

**RESULT:**

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