

SENDING DATA FROM RASPBERRY-PI TO IBM WATSON

Date	7 th November 2022
Team ID	PNT2022TMID53722
Project Name	Smart Waste Management Using IOT in 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

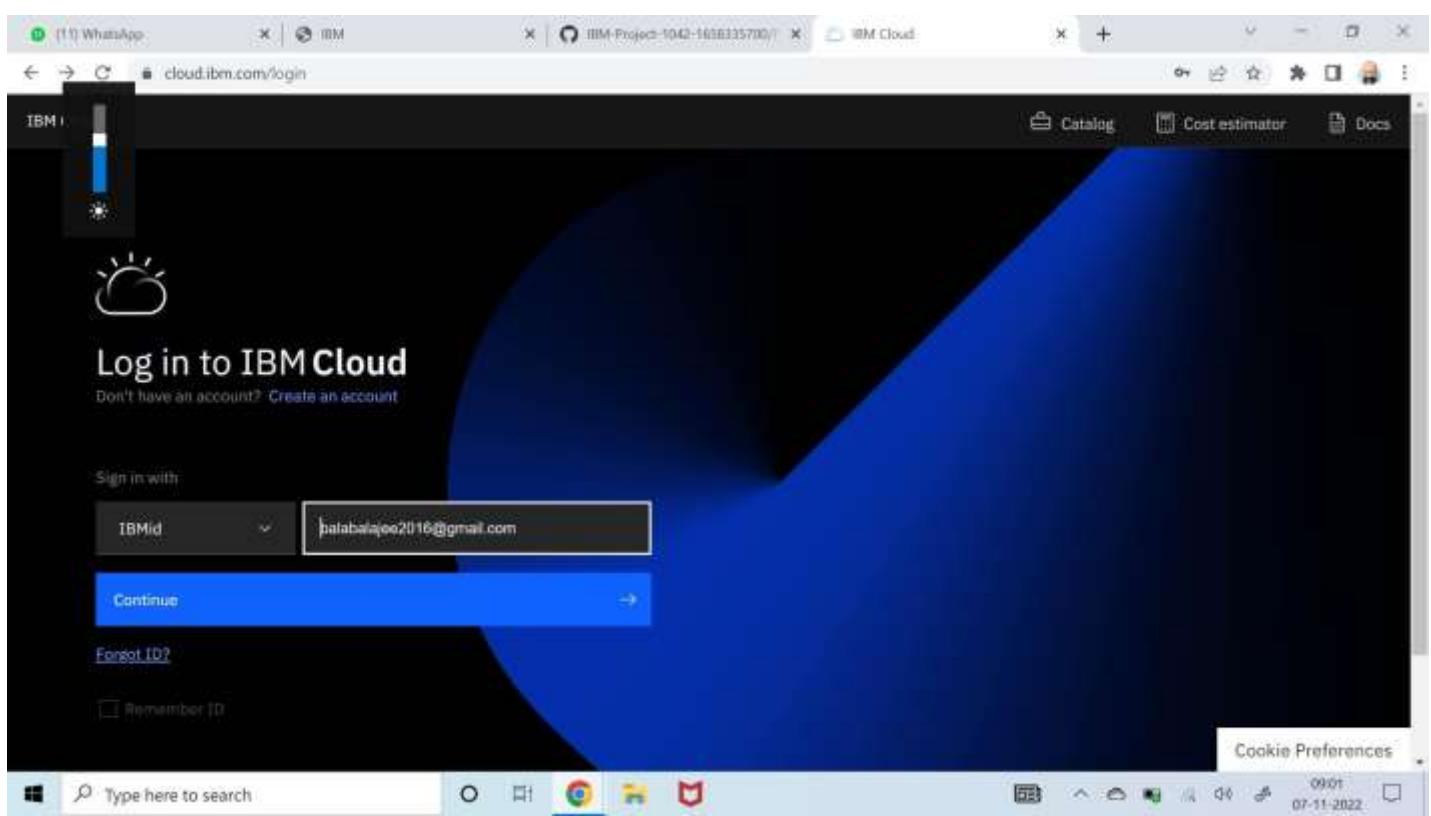
SOFTWARE:

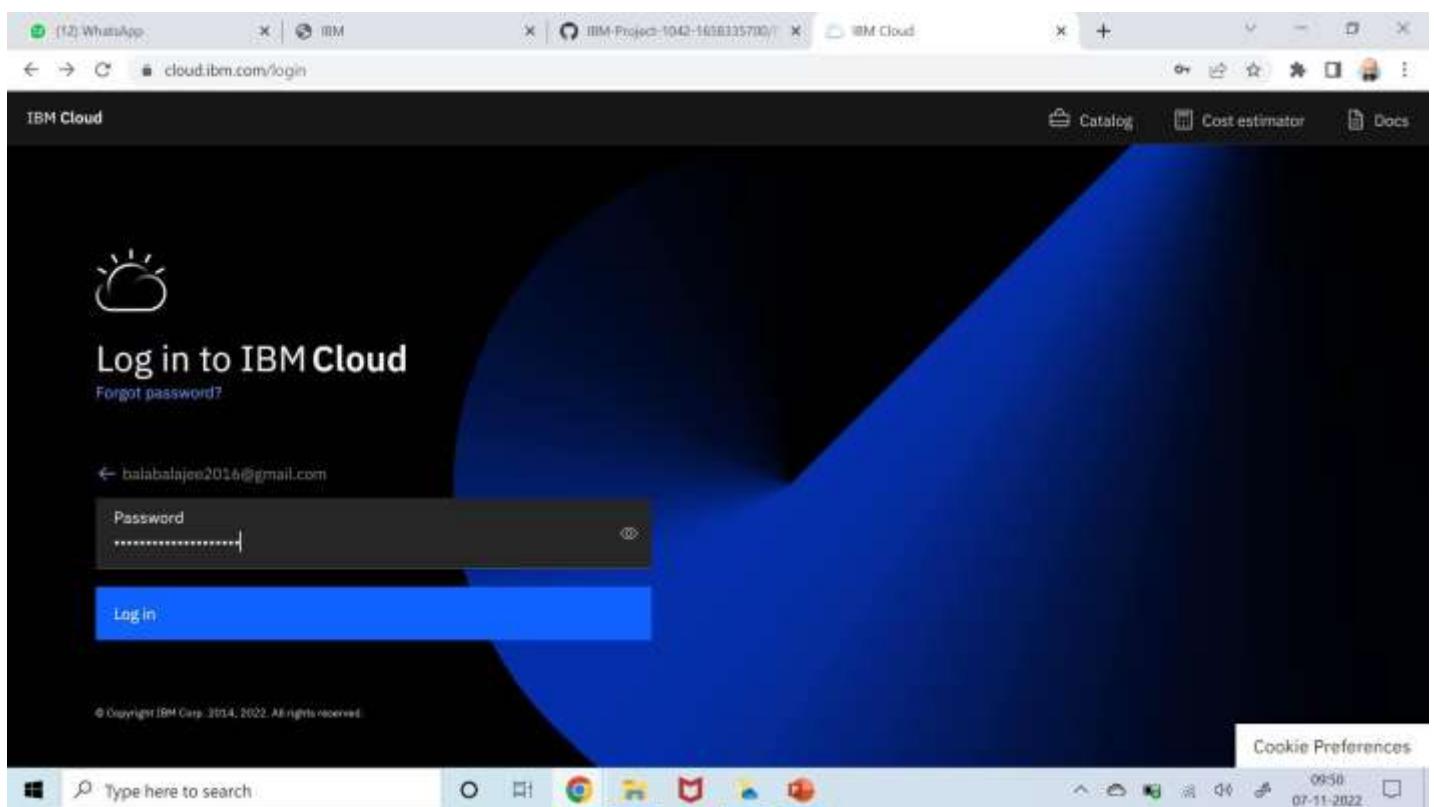
- IBM BLUEMIX ACCOUNT

STEPS TO BE FOLLOWED

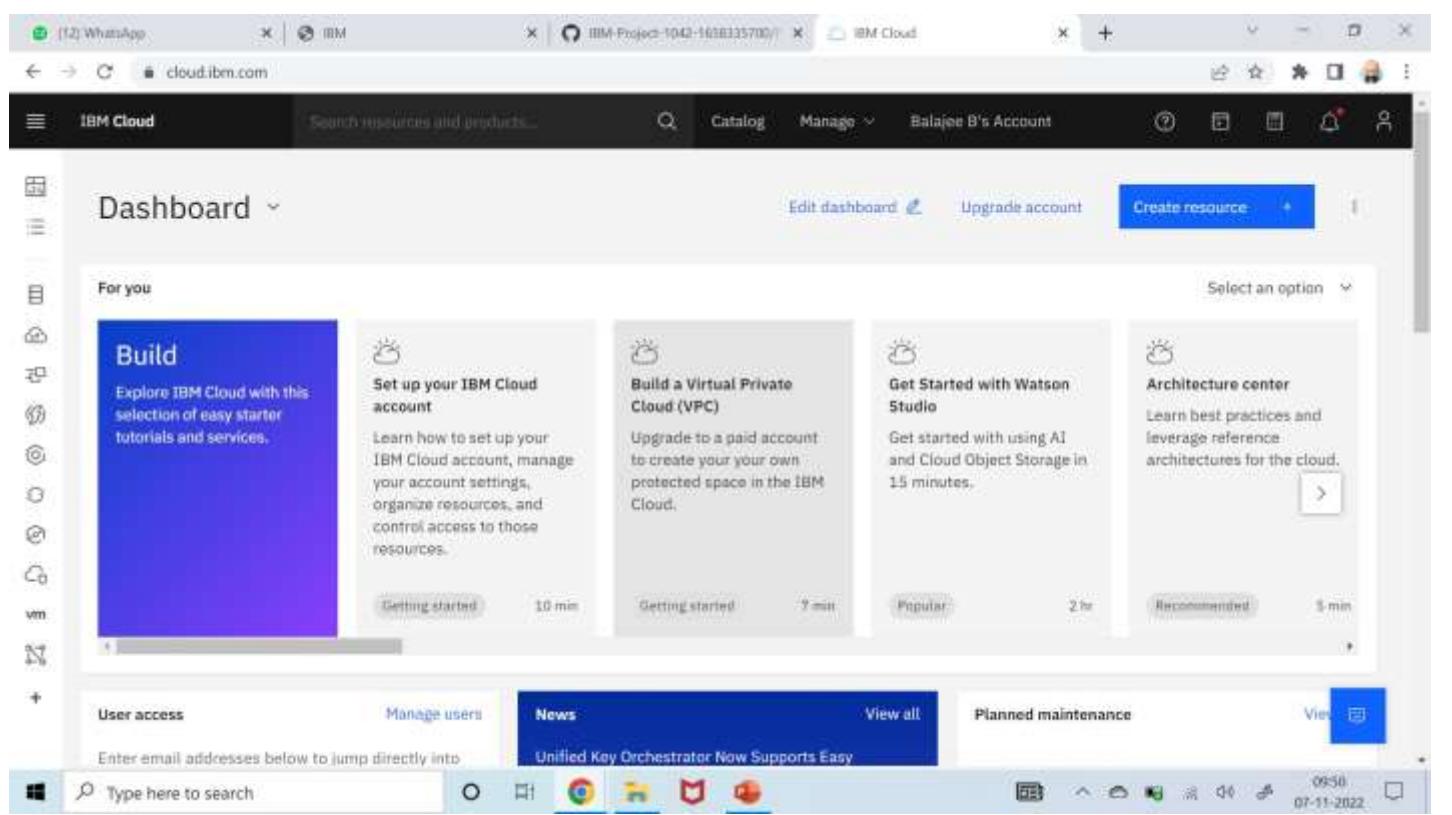
Step-1: Create a device in IBM Watson:

- 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.

The screenshot shows the IBM Cloud Catalog interface. A modal window is open for creating an "Internet of Things Platform". The summary section on the right shows the service is "Free", located in "Frankfurt", and part of the "Internet of Things Platform-og" resource group. The "Create" button is highlighted in blue at the bottom of the modal.

Catalog / 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.

Type: Service Provider: IBM Last updated: 08/15/2022 Category: Internet of Things Compliance: IAM-enabled Location:

Create **About**

Select a location: Frankfurt (eu-de)

Select a pricing plan: I have read and agree to the following license agreements: [Terms](#)

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	Free

Create **Add to estimate**

The screenshot shows the "Service Details - IBM Cloud" page for the "Internet of Things Platform-og" service. The service is listed as "Active". On the left, there's a sidebar with "Manage" options: Plan, Connections, and a "Let's get started with IBM Watson IoT Platform" section featuring a "Launch" button and "Docs" link. Below this is a "Ready for the next level?" section with "IBM Watson IoT Platform Journey" and "Lite" and "Non-Production" status indicators.

Resource list / Internet of Things Platform-og

Active Add tags: [Edit](#)

Actions...

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 Non-Production

➤ click on Launch

- Dashboard of IBM Watson IoT platform,
- Click on Add device

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.

<input type="checkbox"/>	Device ID	Status	Device Type	Class ID	Date Added

- After click on Add device this page will open

Select a device type for the device that you are adding and give the device a unique ID.

Device Type	Select or create a device type...
Device ID	Enter Device ID

- Go to device type and fill the details.

Add Type

Identity Device Information

Type: Device Or Gateway

Name: Fantastic-4

Description:

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.

- Click on Finish

Add Type

Identity Device Information

Type: Device Or Gateway

Name: Fantastic-4

Description:

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.

- Click on Register Device.

IBM Watson IoT Platform

Register Devices, Define Interfaces

Now that you added a device type, you can register and connect devices for this type.

[Register Devices](#)

Cancel Next

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

IBM Watson IoT Platform

Select a device type for the device that you are adding and give the device a unique ID.

Device Type	Fantastic-4
Device ID	12345

Cancel Next

Browse Devices

All Devices Diagnose

This table shows a summary of all devices that have been added. It can be filtered, customized, and searched on using different

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

➤ Click on Next

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

Serial Number	Enter Serial Number	Manufacturer	Enter Manufacturer
Model	Enter Model	Device Class	Enter Device Class
Description	Enter Description	Firmware Version	Enter Firmware Version
Hardware Version	Enter Hardware Version	Descriptive Location	Enter Descriptive Location

Add Metadata

Back

➤ Click on Next

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 ⓘ

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.

➤ Click on Finish

The screenshot shows the 'IBM Watson IoT Platform' interface for creating a new device. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons for device management. The main content area displays the following information:

- Device Type:** Fantastic-4
- Device ID:** 12345
- View Metadata** button
- Security Token:** To be generated

At the bottom right are 'Back' and 'Finish' buttons. Below this, a section titled 'Browse Devices' lists the newly created device with the ID 12345.

➤ Device is created

The screenshot shows the 'IBM Watson IoT Platform' interface displaying the details of the previously created device. The top navigation bar and sidebar are identical to the previous screen. The main content area shows the device details for '12345':

- Identity:** 12345
- Device Information:** Disconnected
- Fantastic-4:** Device
- Created:** Nov 7, 2022 10:15 AM

A sub-section titled 'Recent Events' shows the live stream of data from the device. The table lists three events:

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

Below the table, a message indicates '1 Simulation running'.

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

- Then open your terminal and type pip install ibmiotf

```
[root@raspberrypi ~]# pip install libmcrypt
Collecting libmcrypt
  Downloading libmcrypt-0.3.6.tar.gz (148K)
    100% |████████████████████████████████| 148.0KB 1.0MB/s
Collecting libcurl=7.1.4 (from libmcrypt)
  Downloading libcurl-7.1.4.tar.gz
Collecting libxml2=2.9.4 (from libmcrypt)
  Downloading libxml2-2.9.4-py3-none-any.whl
    100% |████████████████████████████████| 1.3MB 1.0MB/s
Collecting libyaml=0.1.6 (from libmcrypt)
  Downloading libyaml-0.1.6-py3-none-any.whl
    100% |████████████████████████████████| 1.3MB 1.0MB/s
Collecting pytz=2017.1 (from libmcrypt)
  Using cached pytz-2017.1-py3-none-any.whl
Collecting requests=2.19.1 (from libmcrypt)
  Downloading requests-2.19.1-py3-none-any.whl (33.9KB)
    100% |████████████████████████████████| 33.9KB 1.0MB/s
Collecting rfc3986; python_version >= 3 (from libmcrypt)
  Downloading rfc3986-1.0.0-py3-none-any.whl (54.8KB)
    100% |████████████████████████████████| 54.8KB 1.0MB/s
Collecting valinor=0.10.2 (from libmcrypt)
  Downloading valinor-0.10.2-py3-none-any.whl
Collecting selectio=0.11.0-py3-none-any.whl
Collecting urllib3=1.23.1 (from requests==2.19.1->libmcrypt)
  Downloading urllib3-1.23.1-py3-none-any.whl (12.0KB)
Collecting certifi=2018.4.17 (from requests==2.19.1->libmcrypt)
Collecting idna=2.8 (from requests==2.19.1->libmcrypt)
  Downloading idna-2.8-py3-none-any.whl (50.0KB)
    100% |████████████████████████████████| 50.0KB 1.0MB/s
Collecting phonenumbers=1.9.0 (from requests==2.19.1->libmcrypt)
  Downloading phonenumbers-1.9.0-py3-none-any.whl (1.4MB)
    100% |████████████████████████████████| 1.4MB 1.0MB/s
Collecting certifi=2017.4.17 (from requests==2.19.1->libmcrypt)
  Using cached certifi-2017.4.17-py3-none-any.whl
Building wheels for collected packages: libmcrypt, idna, phonenumbers
  Running setup.py bdist_wheel for libmcrypt ... done
  Stored in directory: /home/pi/.cache/pip/wheels/5a/7f/45/0be32a887e62ff7bd6e6c0d1d12a2031/libmcrypt-0.3.6
  Running setup.py bdist_wheel for idna ... done
  Stored in directory: /home/pi/.cache/pip/wheels/0a/42/66/070011679c77971344e7f88477bd3e6613cf7f19483
  Successfully built libmcrypt idna
  Installing collected packages: idna, phonenumbers, libmcrypt, requests, phonenumbers, idna
Successfully installed idna-2.8.1 pytz-2017.2 requests-2.19.1 phonenumbers-1.9.0 libmcrypt-0.3.6 certifi-2017.4.17 phonenumbers-1.9.0
[113.0]
```

- 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
#Import required packages
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "MyPiDev",
        "typeId": "NodeMCU",
        "deviceId": "12345"
    },
    "auth": {
        "token": "12345678"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    myData['command'] = cmd.data['command']

client = wiotp.sdk.device.DeviceClientConfig(myConfig, logHandler=None)
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.

IBM Watson IoT Platform

qq4hk.internetofthings.ibmcloud.com/dashboard/devices/browse

IBM Watson IoT Platform

12345 Disconnected Fantastic-4 Device Nov 7, 2022 10:15 AM Add Device

Identity Device Information Recent Events State Logs

The recent events listed show the live stream of data that is coming and going from this device.

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

1 Simulation running

ibm.csv Show all

Type here to search

Nov 7, 2022 10:15 AM

1 Simulation running

ibm.csv Show all

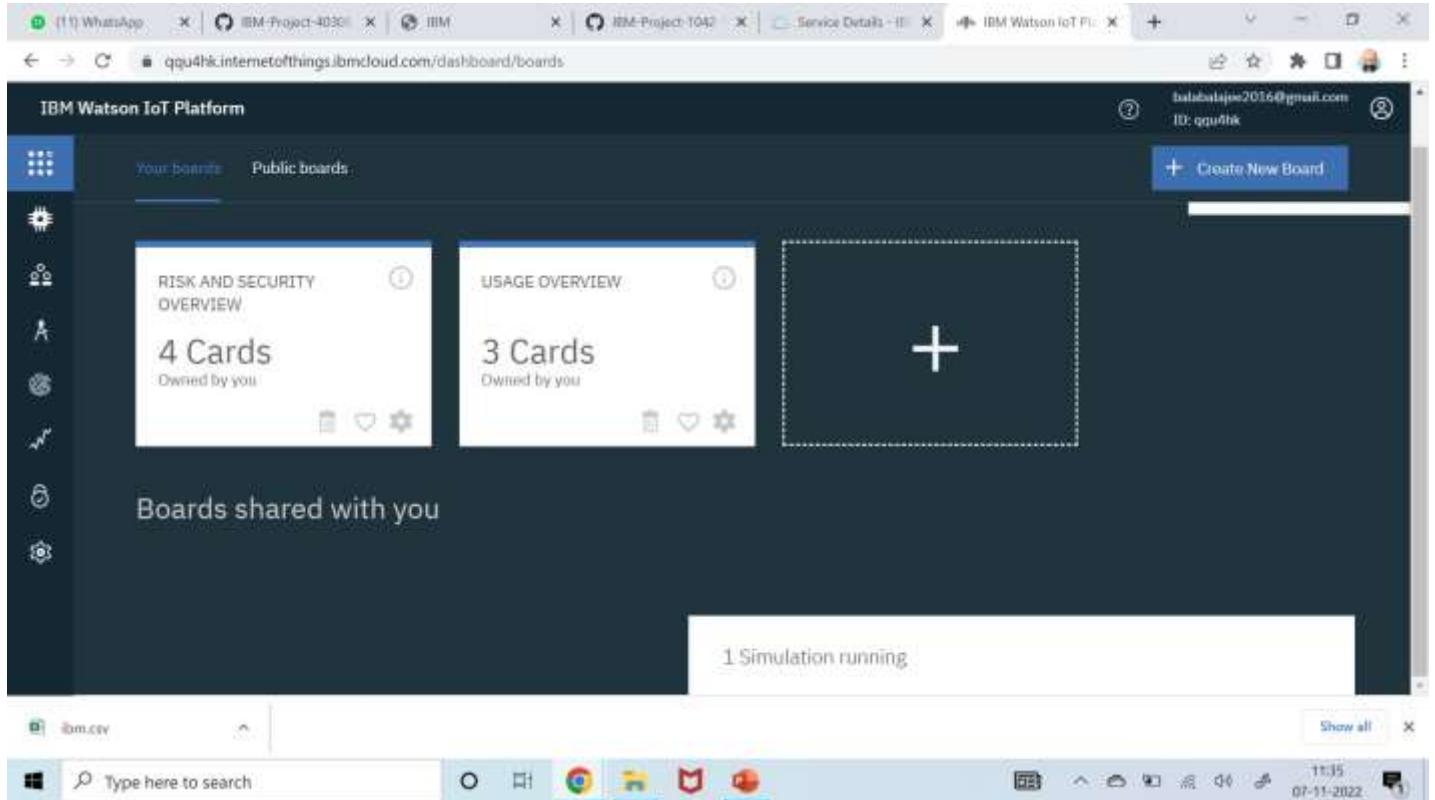
Type here to search

Nov 7, 2022 10:15 AM

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

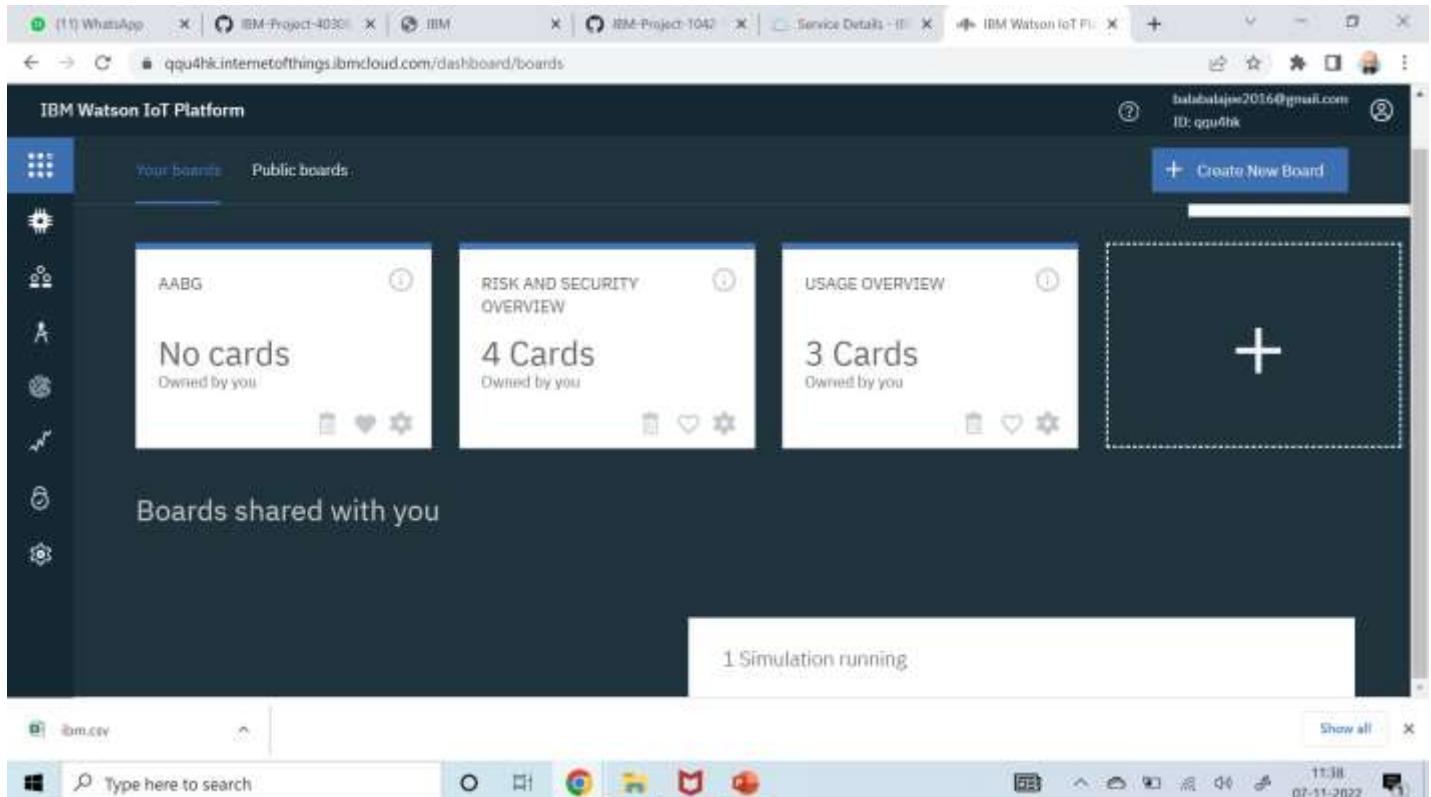
Step-4: Creating boards and cards for visualization of data:

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

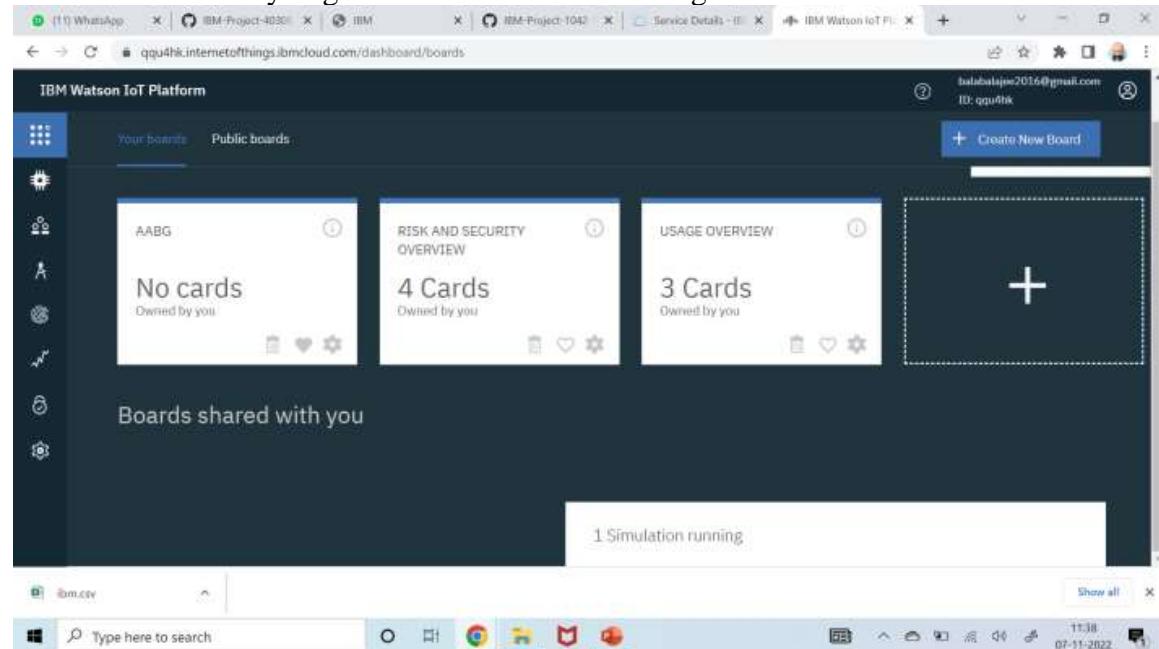


- Click on Create a new board to create a board .

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

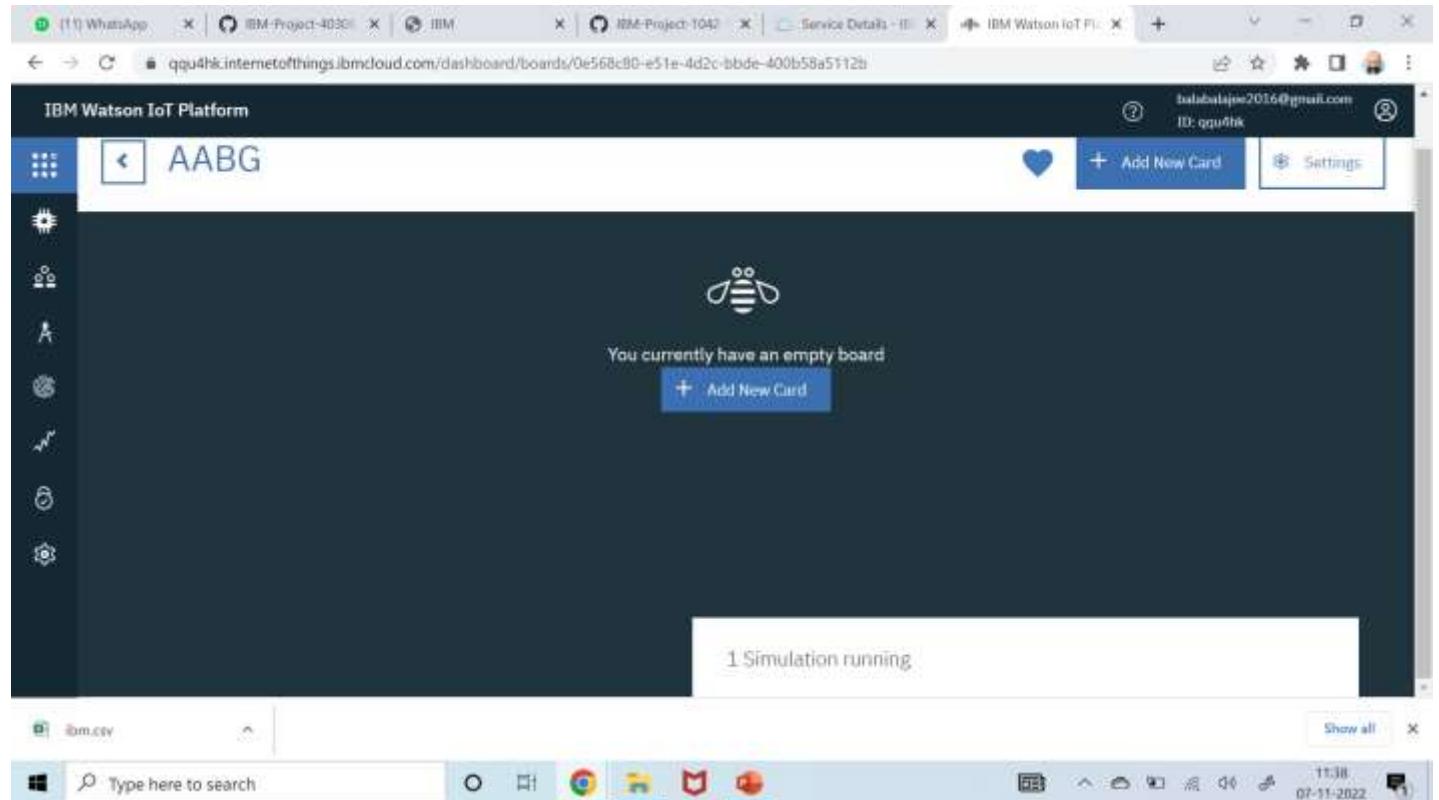


- Then click on Next you get the below window then again click on Sub

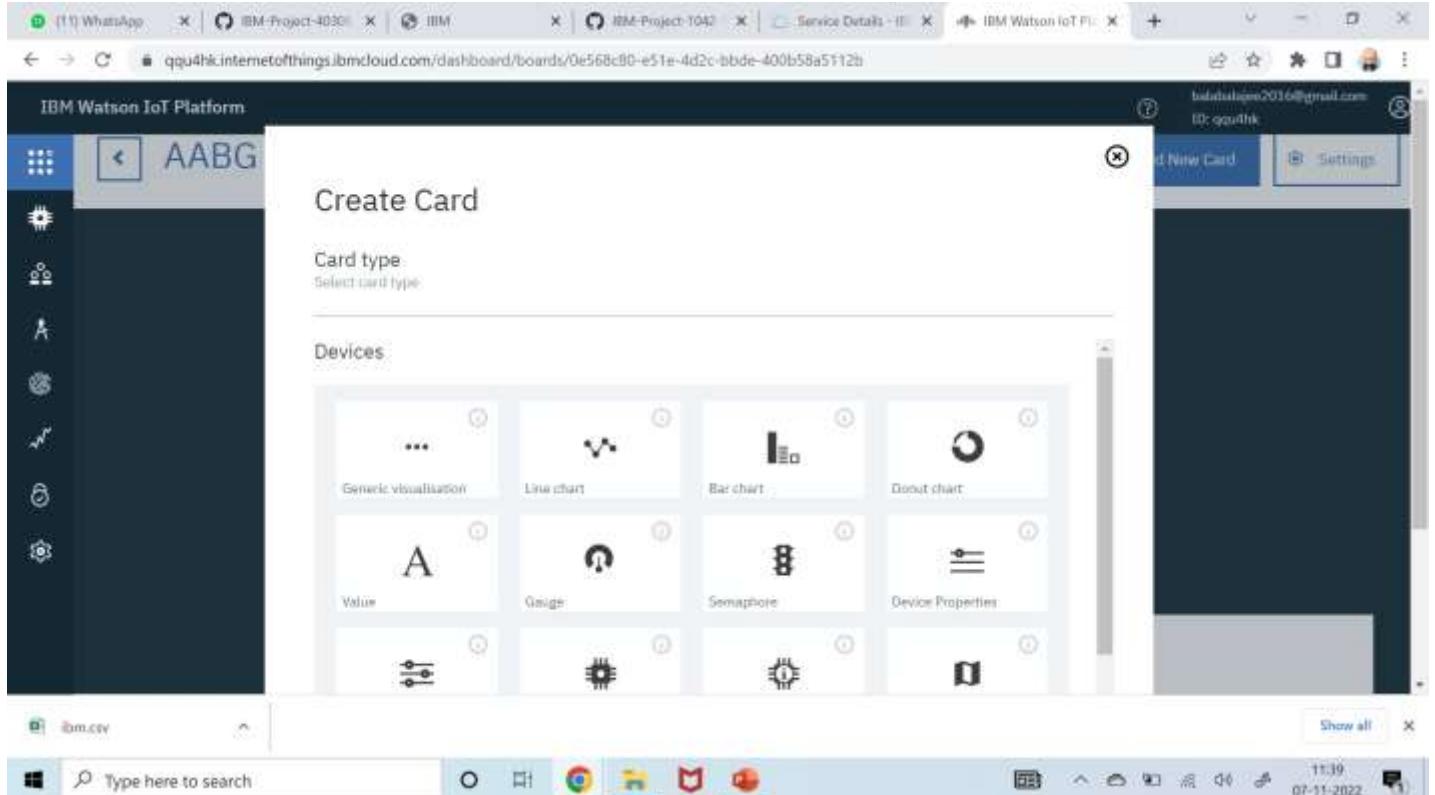


Then double click on your boards name which you have created.

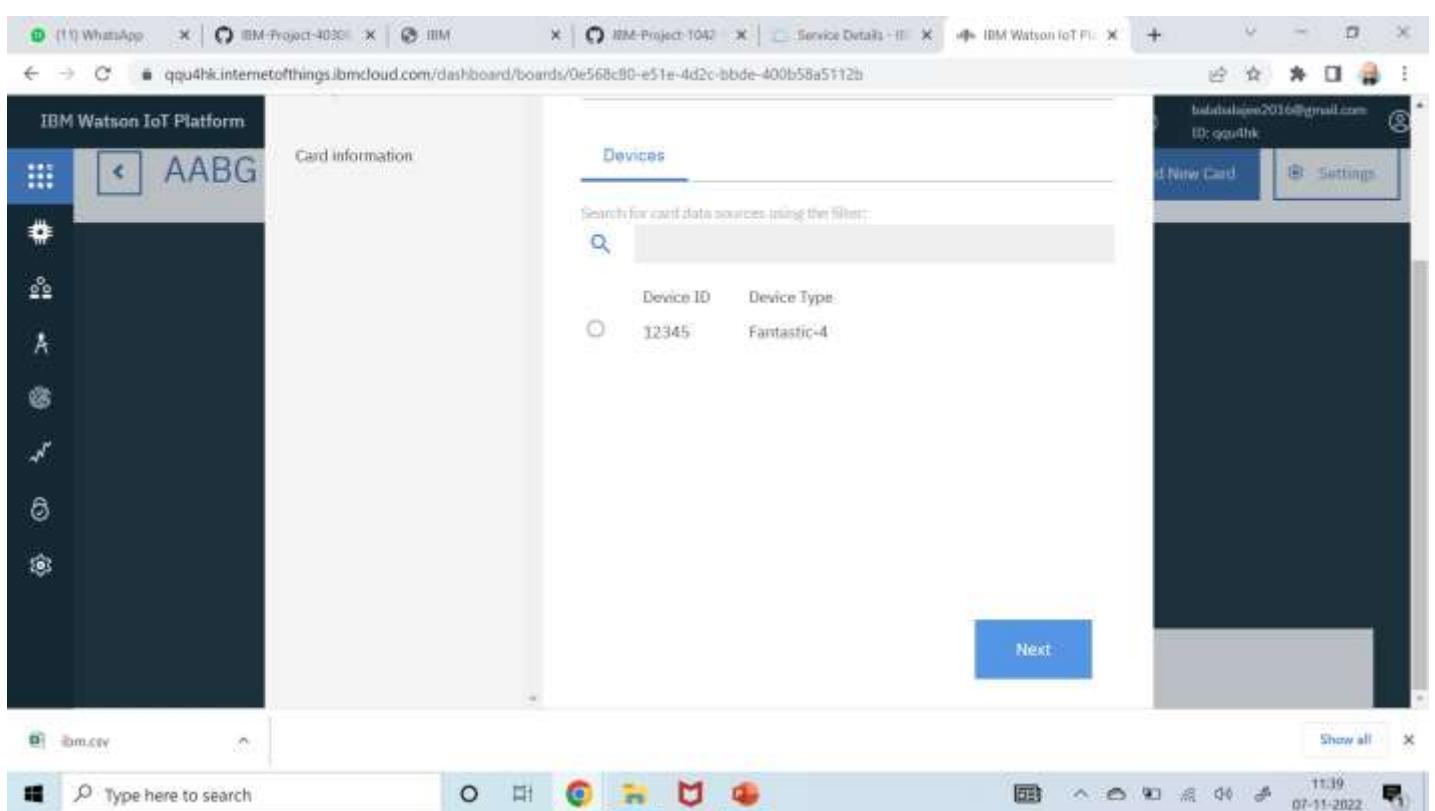
- Click on Add New Card



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

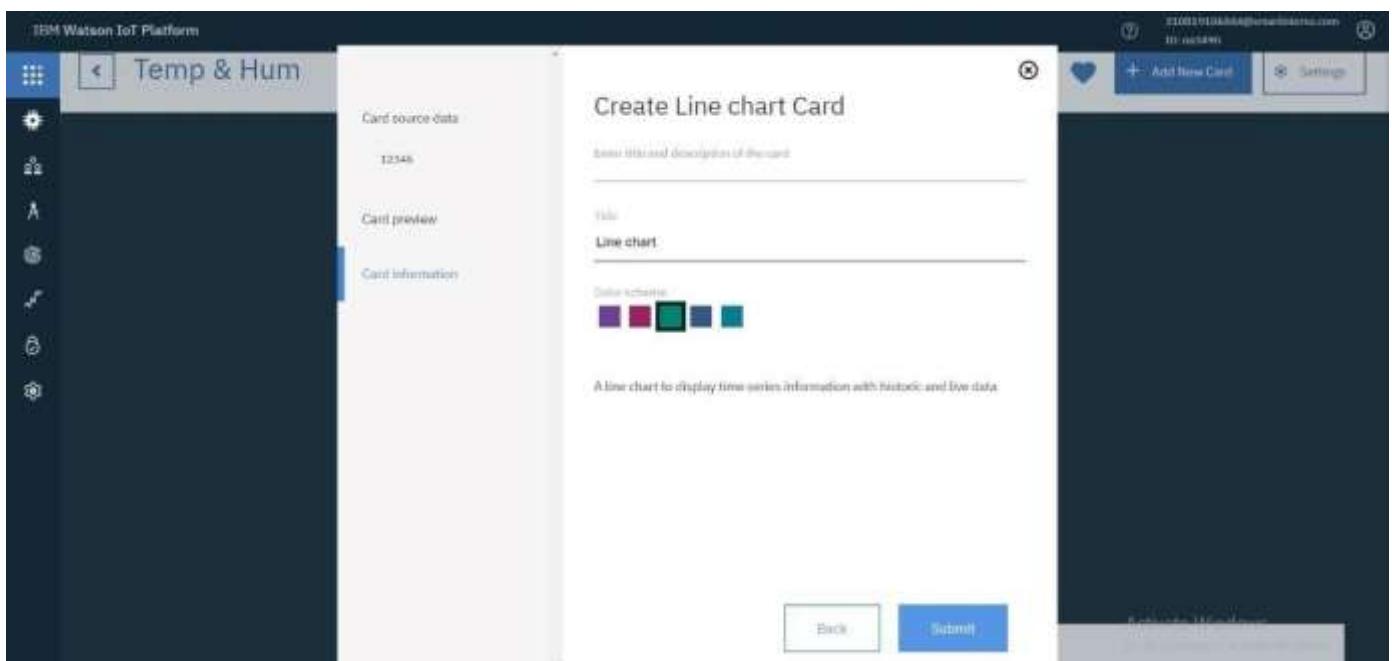


- You get the below window, choose the Device and click on Next.

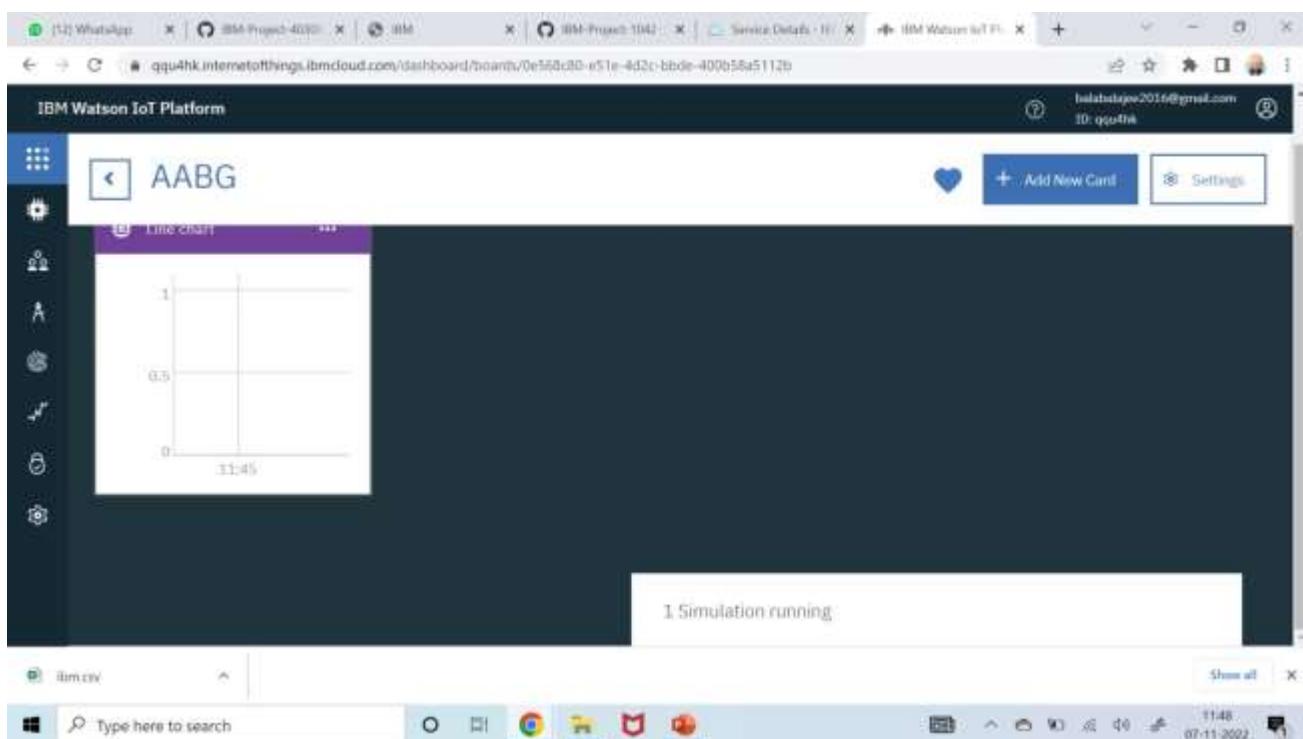


- Select the event, properly to be visualized on your graph and click next. In my case it is humidity

- 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.