

SENDING DATA FROM RASPBERRY-PI TO IBM WATSON

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

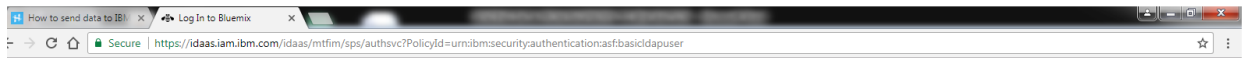
SOFTWARE:

- IBM BLUEMIX ACCOUNT

STEPS INVOLVED:

Step-1: Create a device in IBM Watson:

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

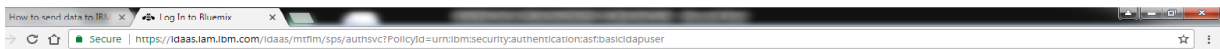


Log into IBM Bluemix

Enter Email or IBMid: [Forgot your IBMid?](#)

Continue

New to Bluemix? [Sign Up](#)



Log into IBM Bluemix

IBMid: nisanthtechnovators@gmail.com

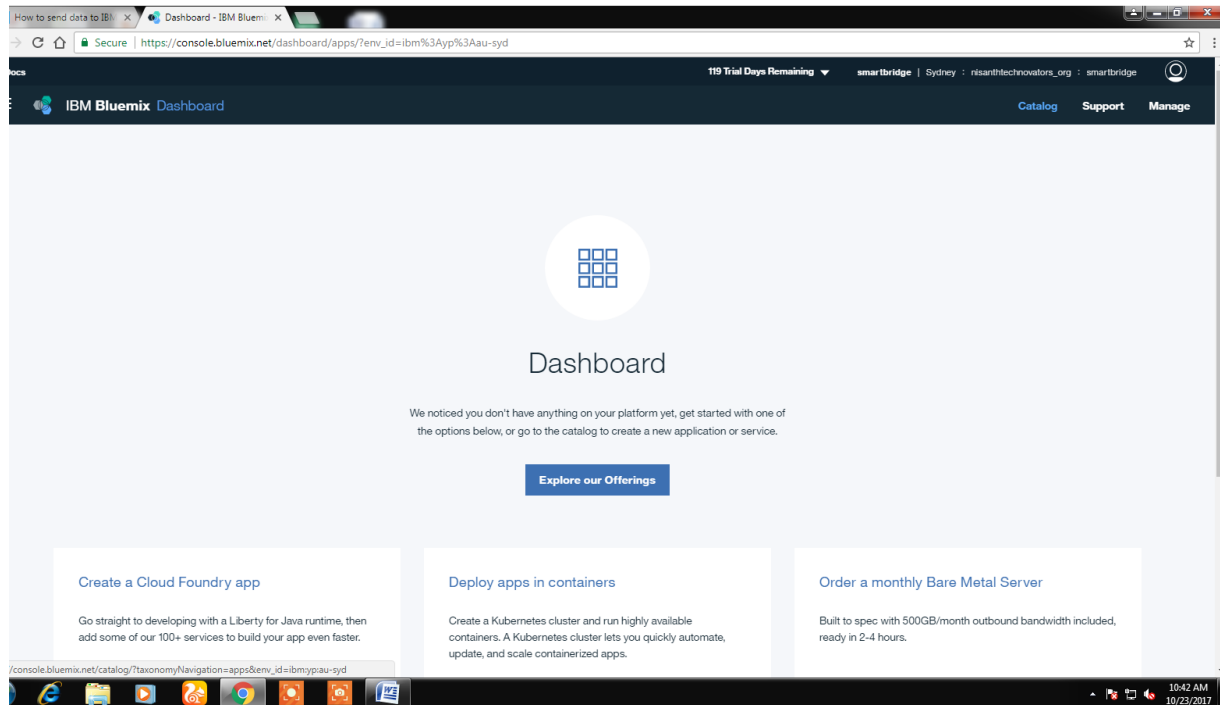
Password [Forgot your password?](#)

Log in

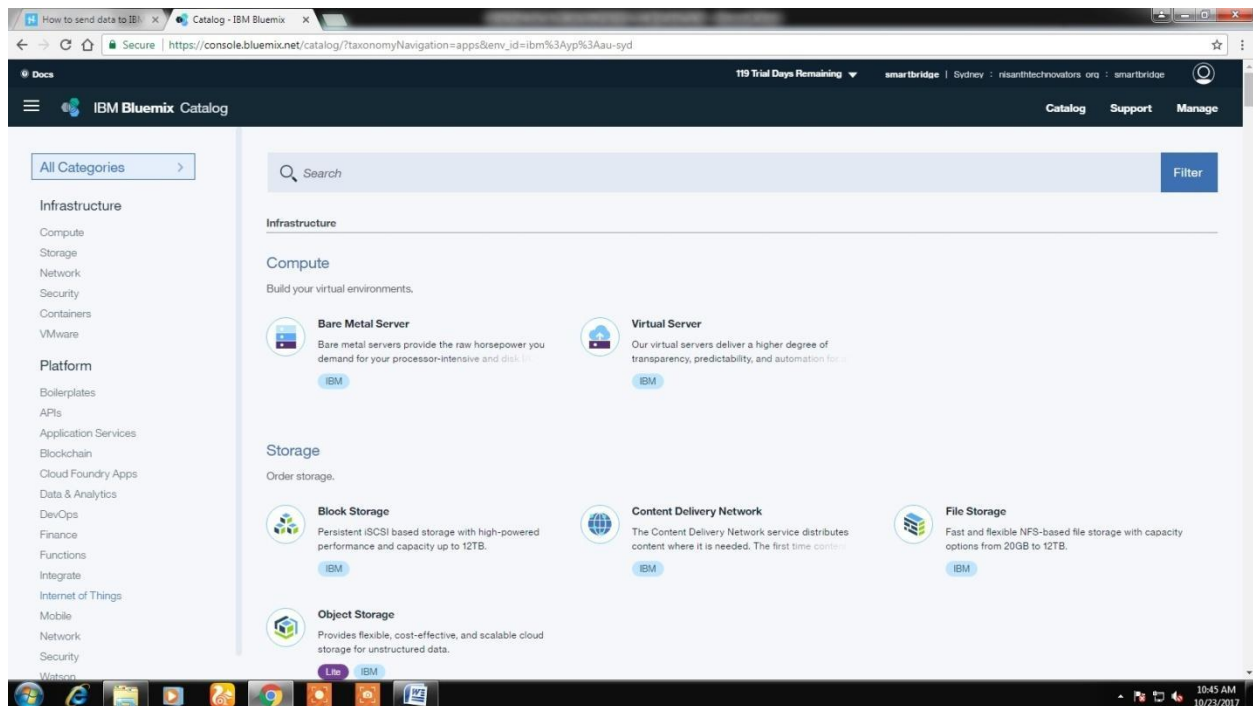
[Use a different IBMid or email](#)



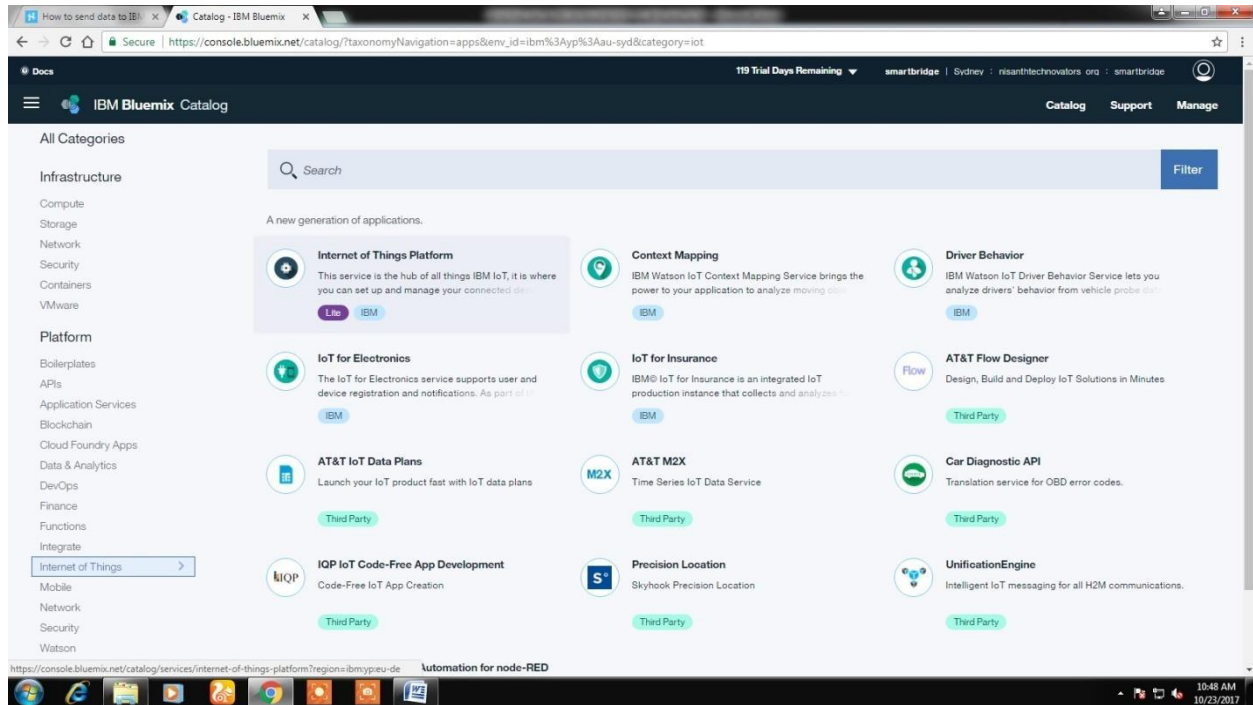
- Click on catalog on your dashboard screen



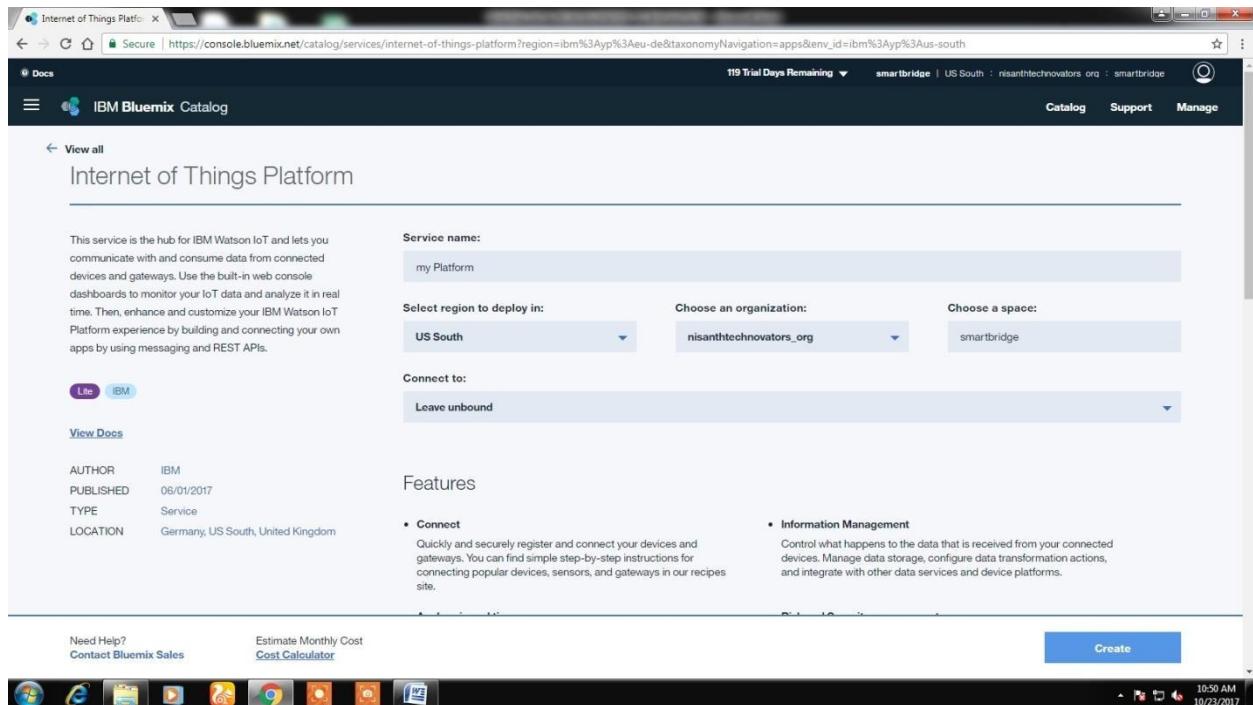
- Under Platforms Go to Internet of Things.



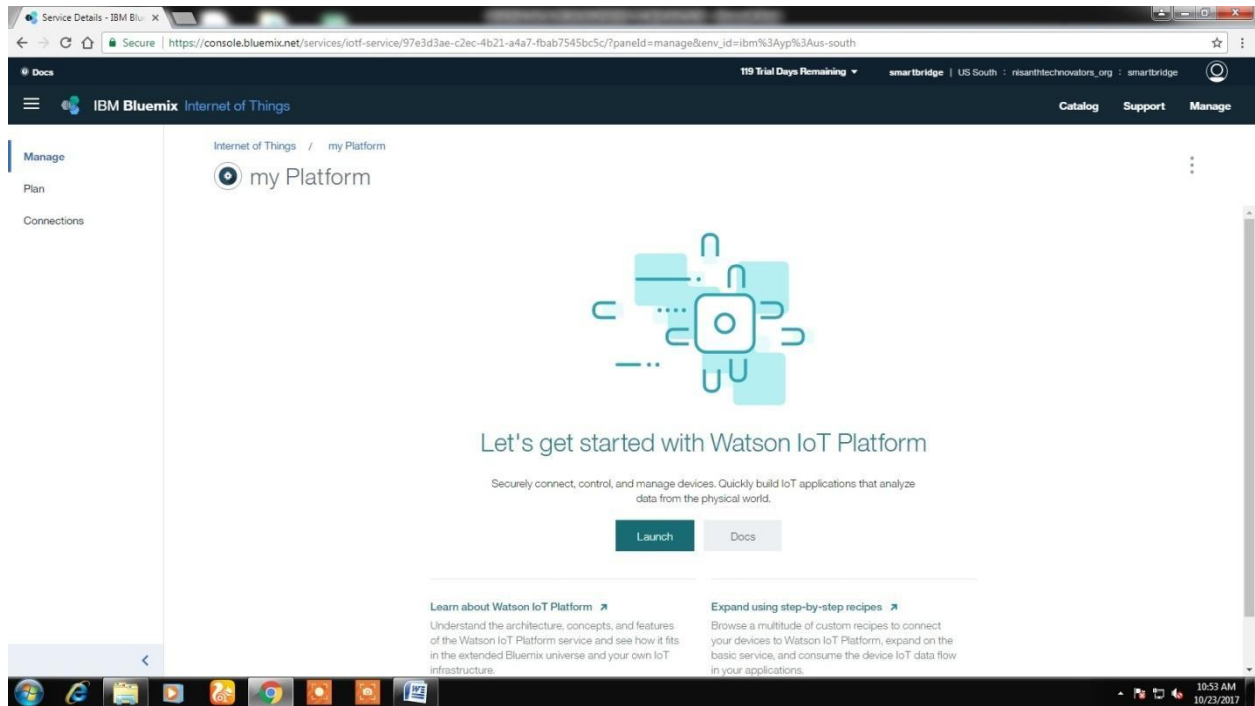
- Under Internet of Things Internet of Things Platform



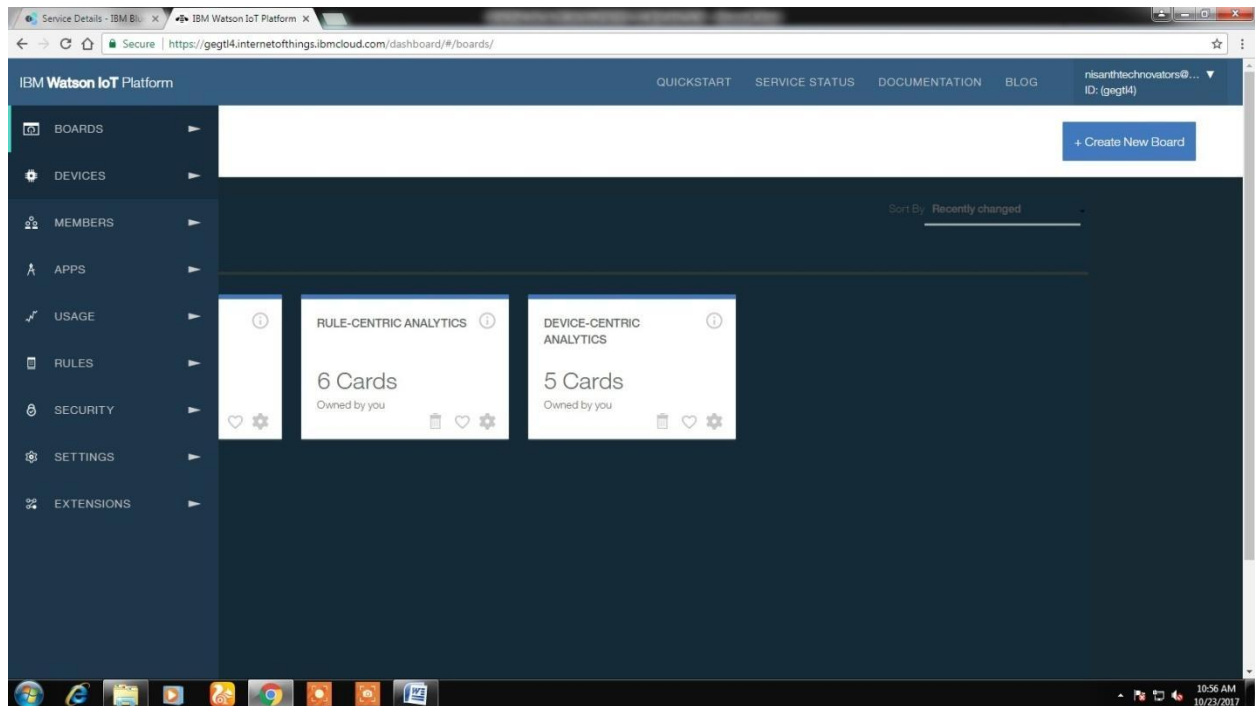
- Then give a name for your Service name and click on Create



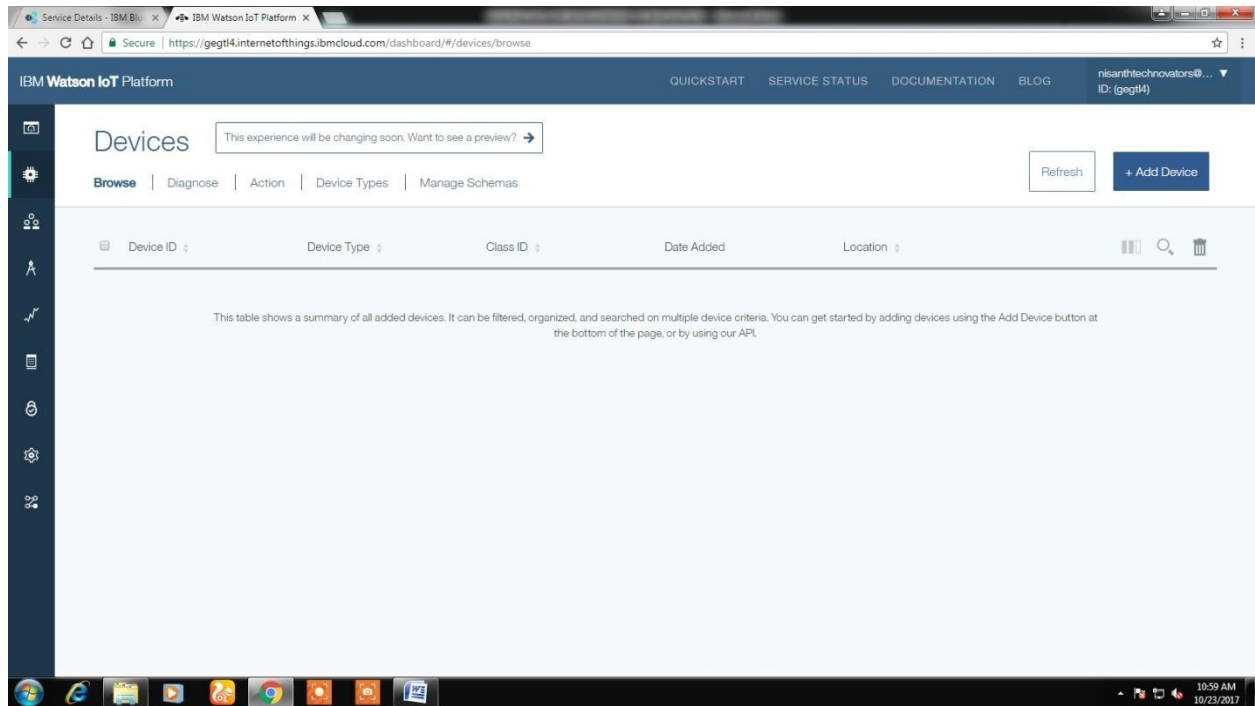
- After getting into your service click on Launch



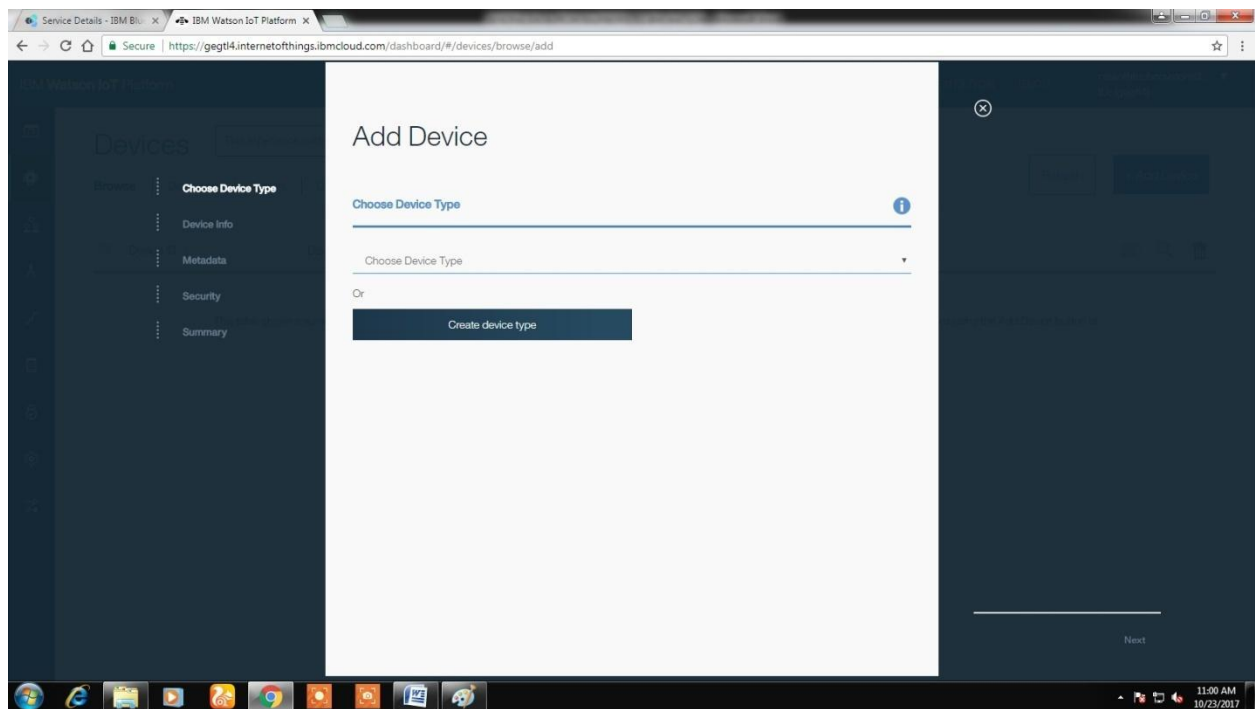
- Then you get into IBM Watson Platform .Then click on Devices



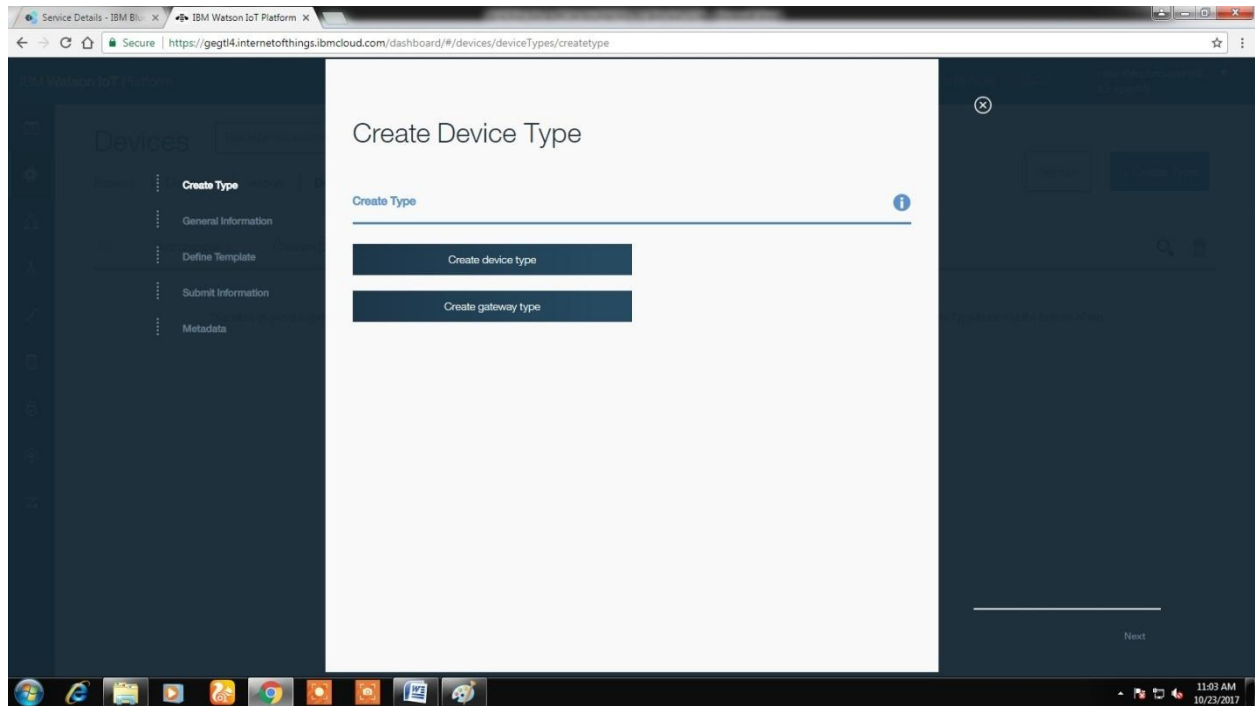
- When you get into Devices you find a button called +Add Device click on it.



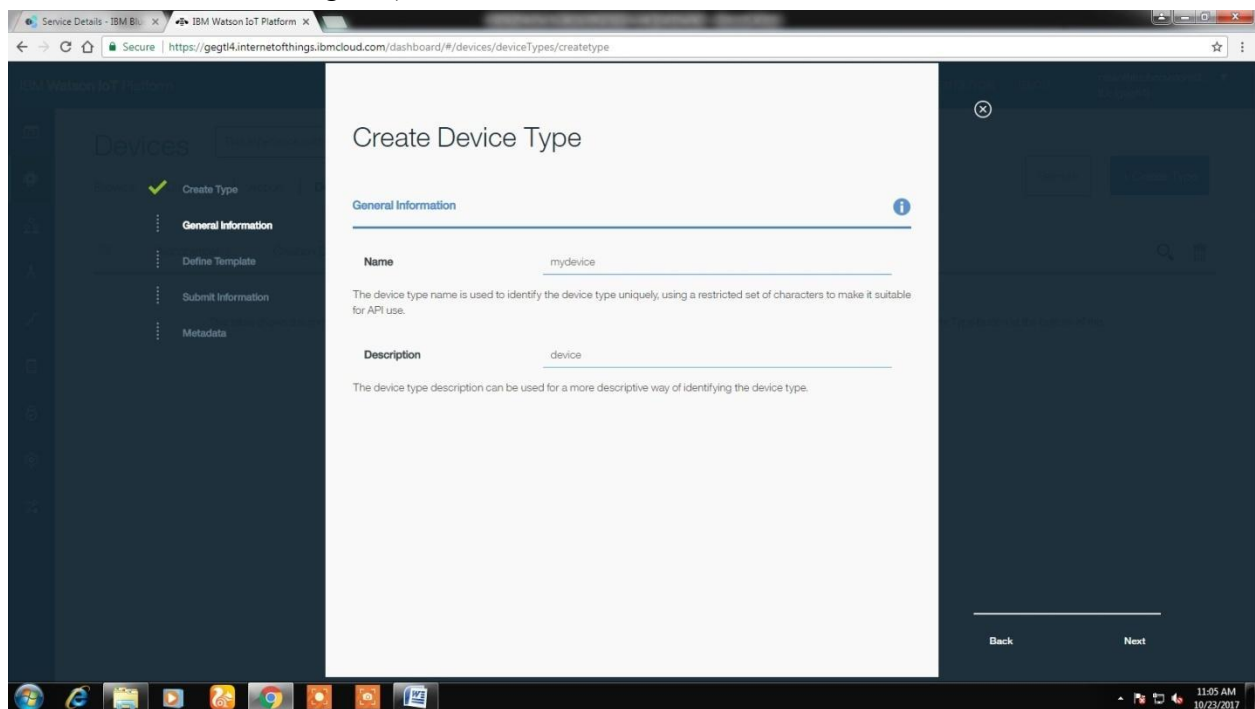
- Then you get a window where you should click on Create Device Type.



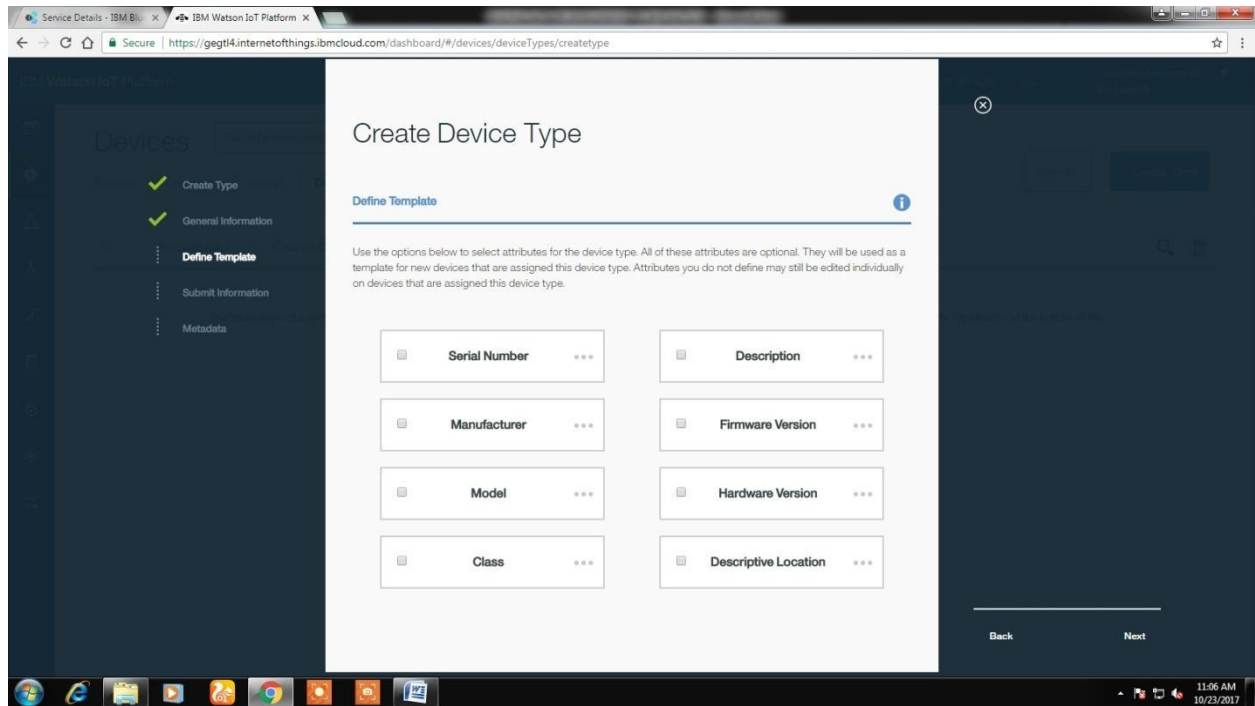
- Then you will get an another window asking whether to create a device type or gateway type. Click on “Create a Device Type”



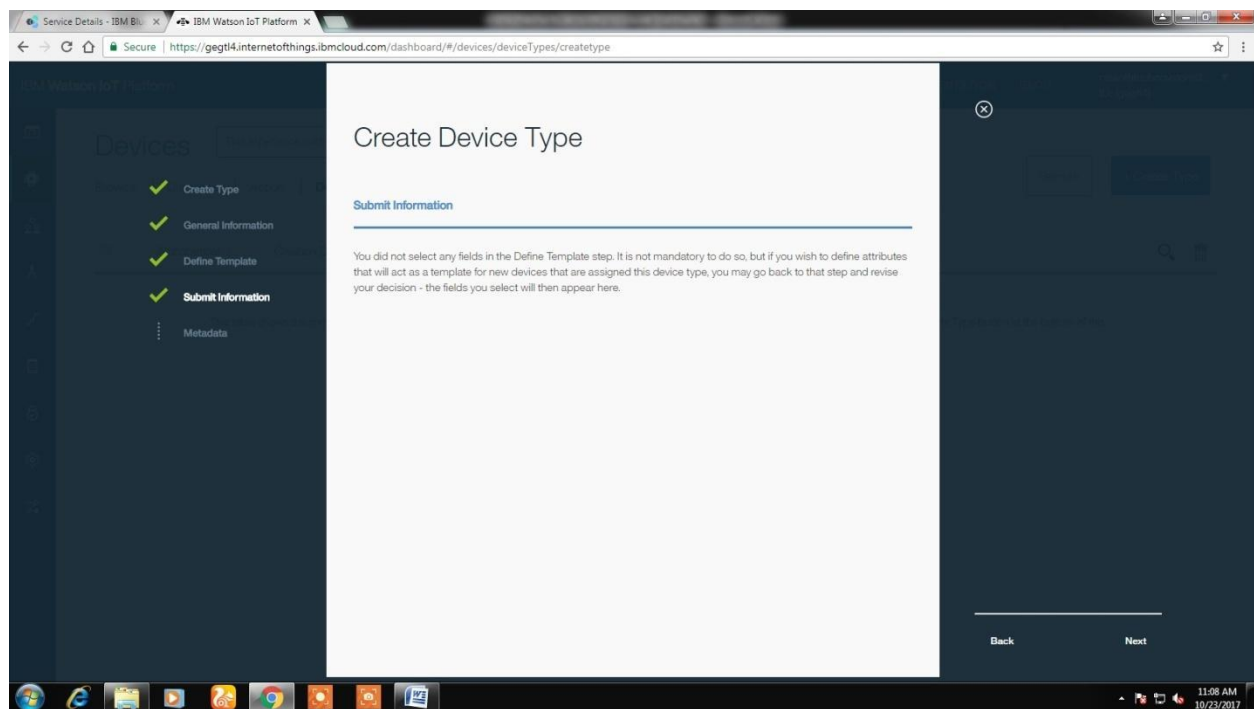
- Then give a name and Description for your device (any name and description can be given) and then click on Next



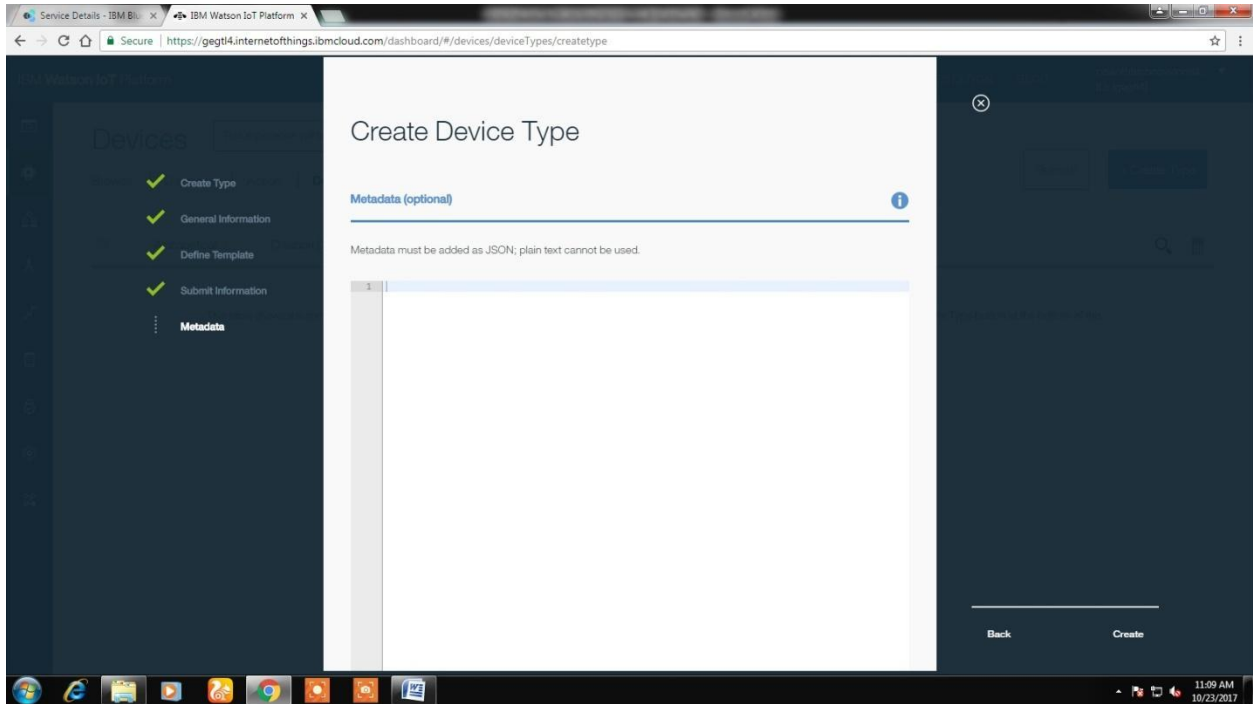
- Then another Window appears just click on Next



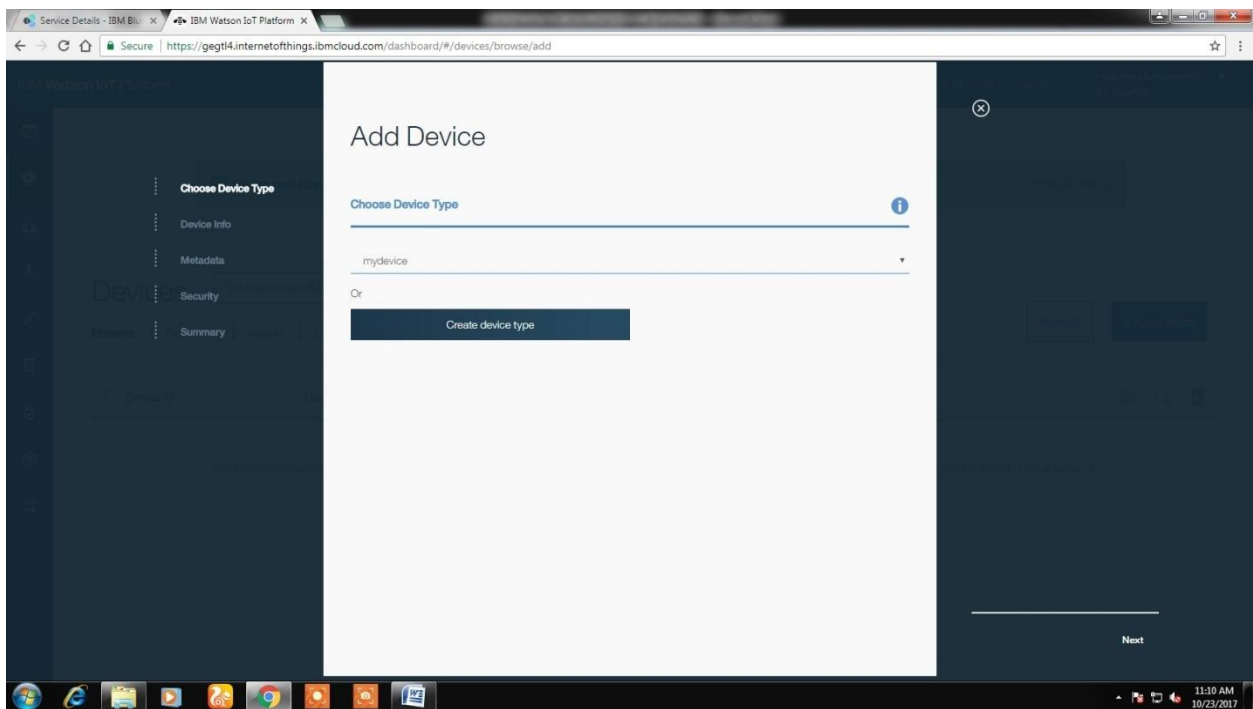
- Click on Next for the above window



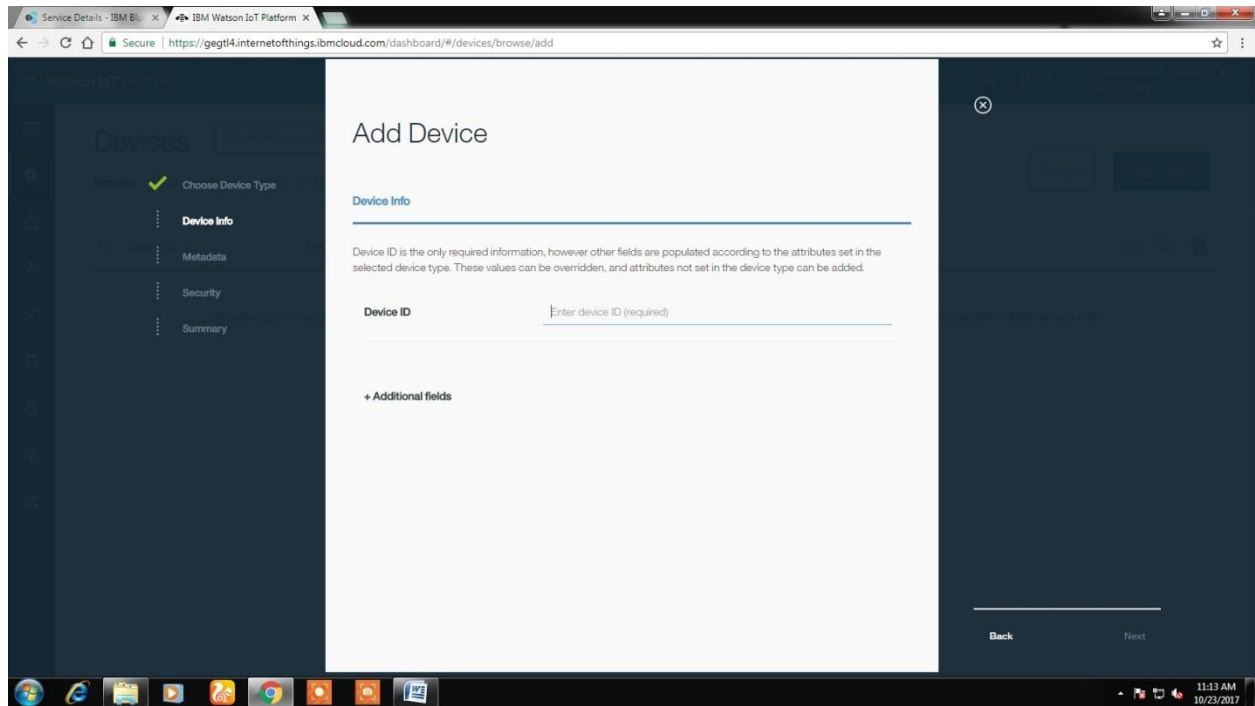
- Click on Next for the above window



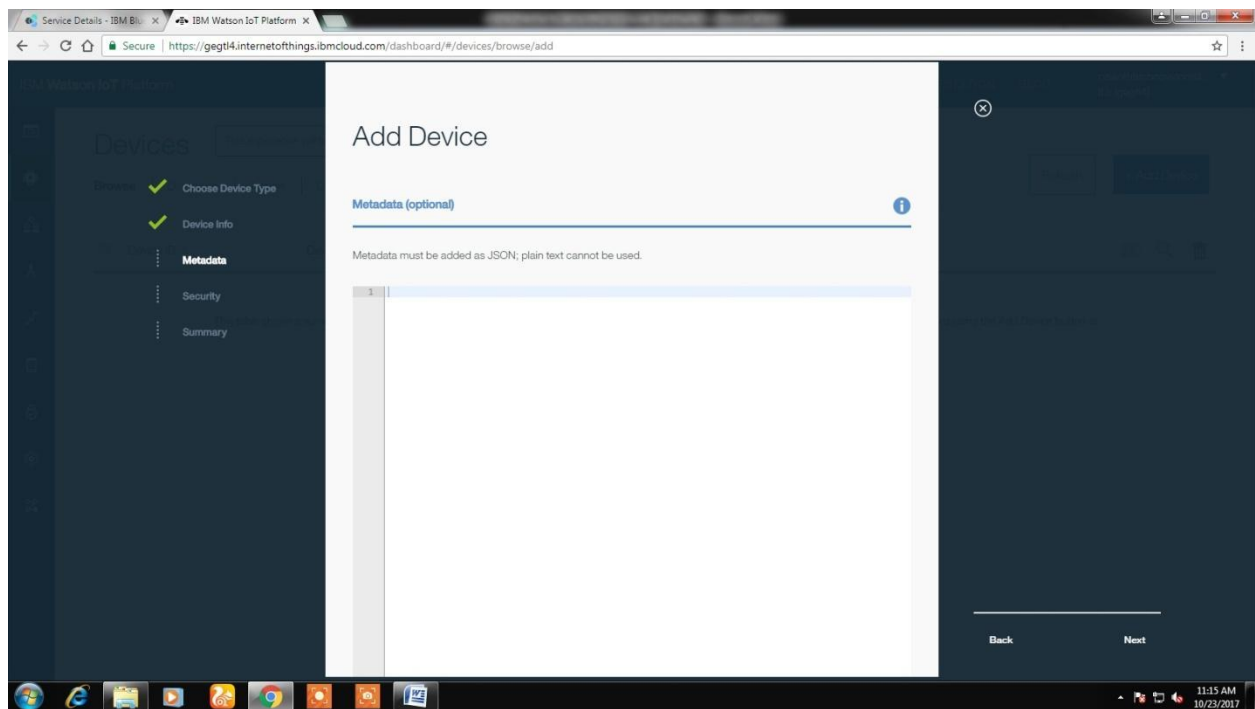
- Click on Create at the bottom when the above window appears.



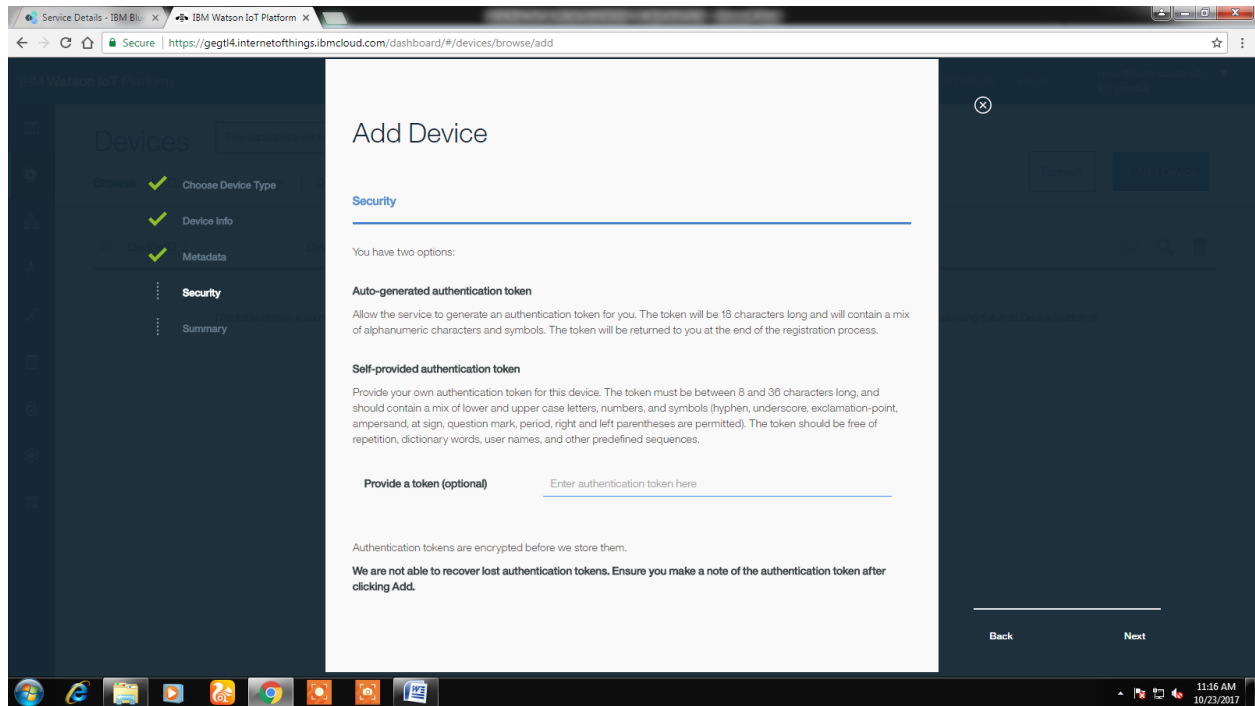
- Then click on Next when the above window appears.



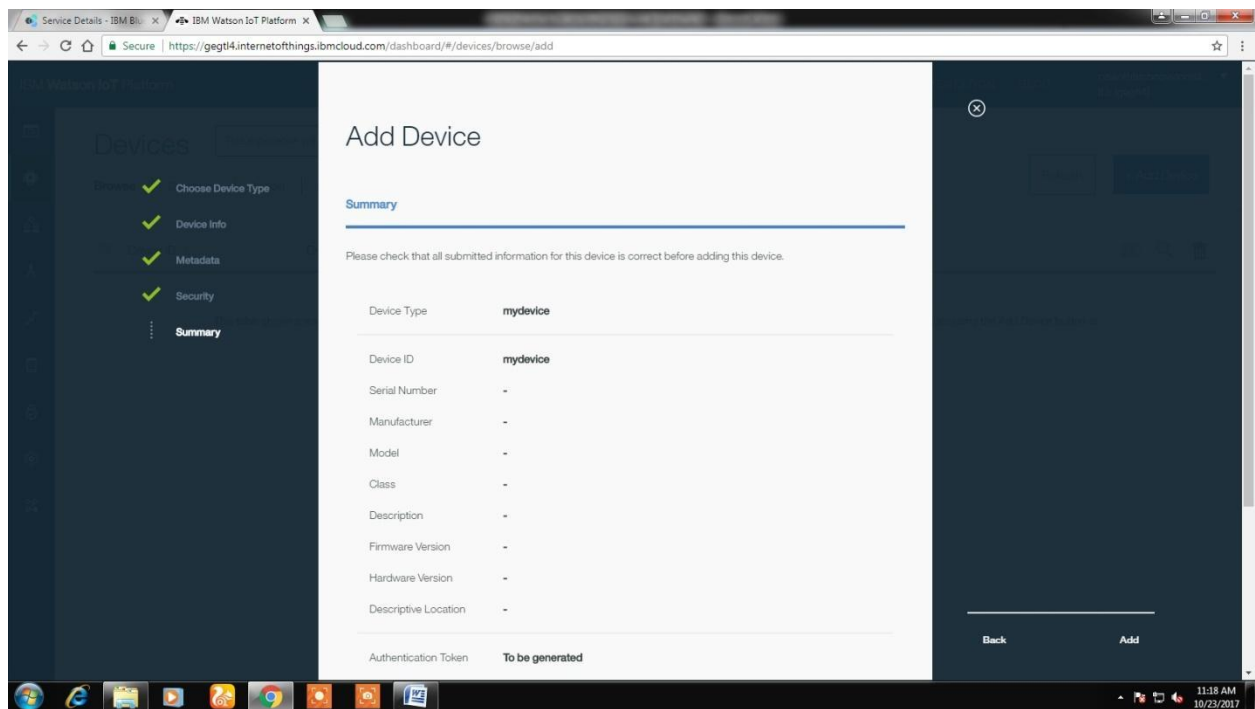
- Give your Raspberry-Pi's IP Address as Device ID for unique identification



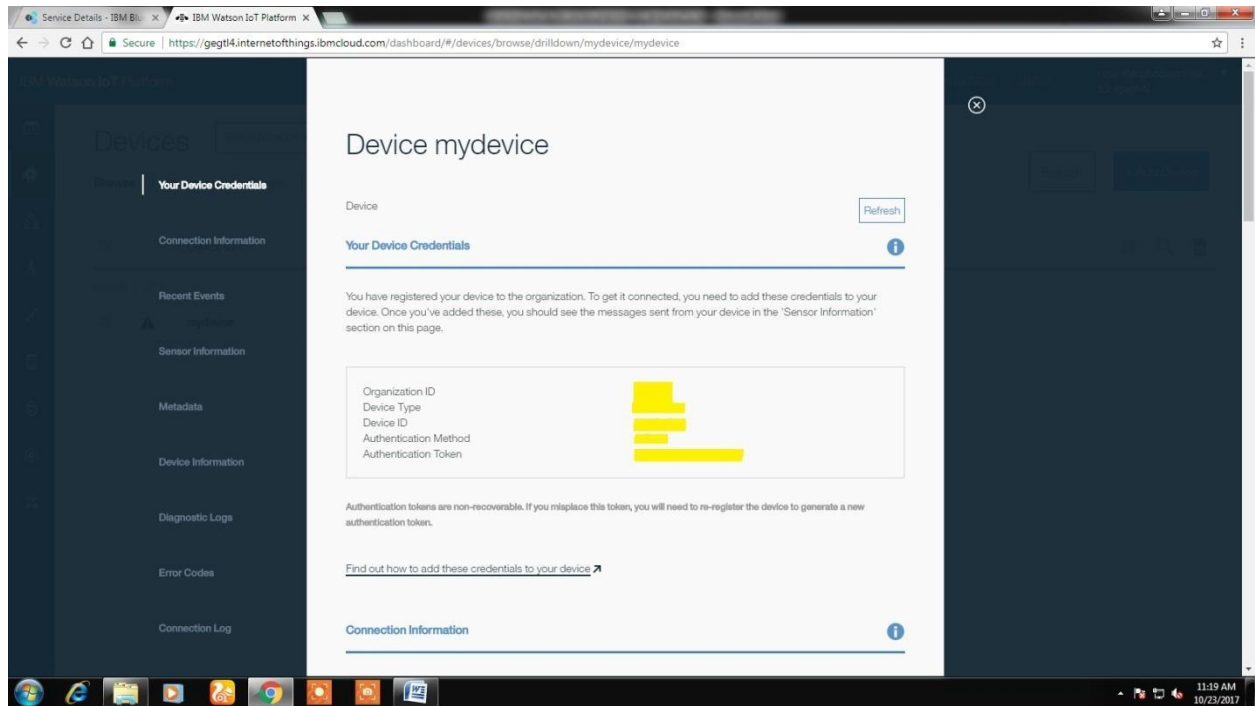
- Click next when the above window appears.



- Again click on Next when the above window appears.



- Then click on Add when the above window appears.



- Then you get your device Credentials which you can use later. Copy them and paste them in a notepad for future uses.

STEP-2: INSTALLING NECESSARY PACKAGES ON YOUR PI:

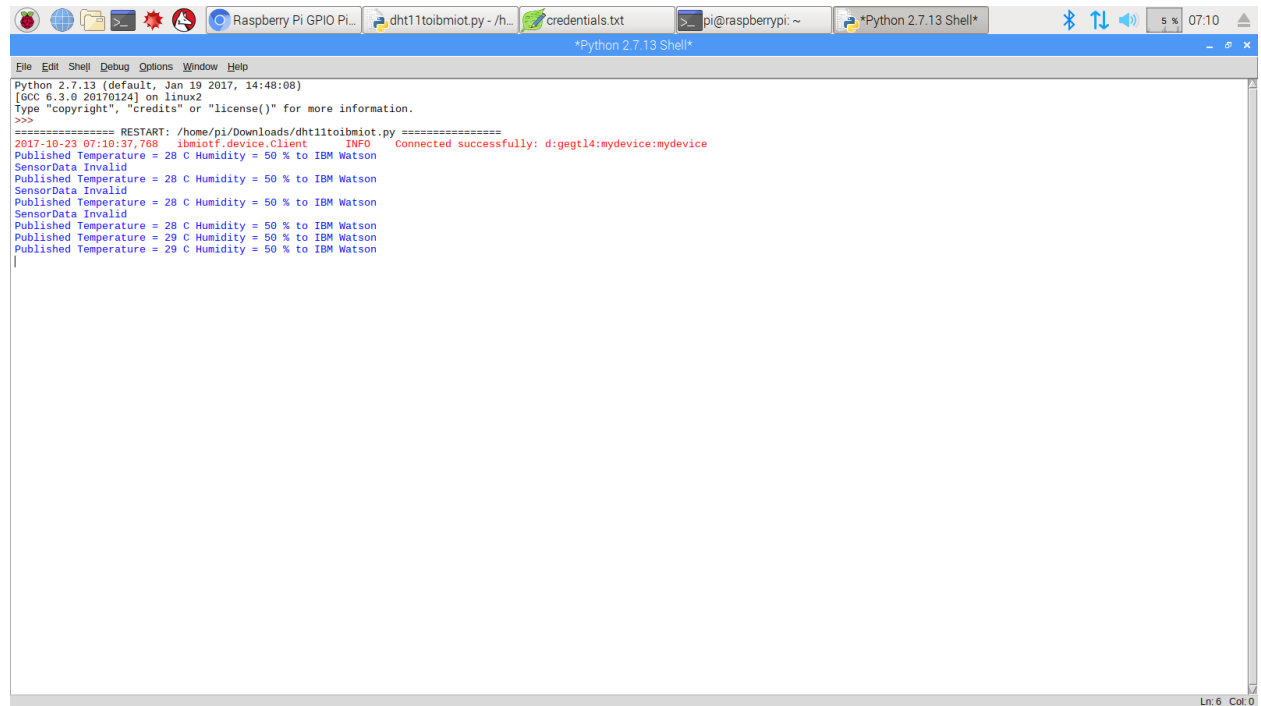
- Now we are going to install necessary packages on your pi.
 - Open your terminal in your pi and type the following commands
 - `curl -LO https://github.com/ibm-messaging/iot-raspberrypi/releases/download/1.0.2.1/iot_1.0-2_armhf.deb`
 - `sudo dpkg -i iot_1.0-2_armhf.deb`
 - `service iot status`
- Following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

```
pi@raspberrypi: ~  
File Edit Tabs Help  
2017-10-23 06:55:22: http://ftp.nl.debian.org/debian/pool/main/o/openssl/lib  
s11.0.0.1.0.1t-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  
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-sys-generator(8)  
CGroup: /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[2557]: 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.  
Oct 23 06:56:25 raspberrypi iot[2562]: Connecting to tcp://quickstart.messaging.internetofthings.ibmcloud.com:1883 with client Id: d:quickstart:iot-sample-raspberrypi:b827eb8c6734  
Oct 23 06:56:25 raspberrypi iot[2562]: Connection was successful  
pi@raspberrypi:~$ scot  
bash: scot: command not found  
pi@raspberrypi:~$
```

- Then open your terminal and type pip install ibmiotf

```
pi@raspberrypi: ~  
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 (80kB)  
    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>=0.7.0 (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-0.11.0-py2.py3-none-any.whl  
Collecting urllib3<2.2,>=1.21.1 (from requests>=2.5.0->ibmiotf)  
  Downloading urllib3-1.22-py2.py3-none-any.whl (132kB)  
    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 (54kB)  
    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 1.6MB/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/f9/45/bbc33ad957e82f7b71ba80e316d65a83d9d735a0d12e0c0418  
Running setup.py bdist_wheel for dicttoxml ... done  
Stored in directory: /home/pi/.cache/pip/wheels/45/62/59/9e910b3ec6a7b2ae66a13765401b50def5468024078e12cce  
Running setup.py bdist_wheel for paho-mqtt ... done  
Stored in directory: /home/pi/.cache/pip/wheels/20/d8/0d/acdc8f289011b7be7de71deebe0642fb83be0313dfff0493  
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:~$
```

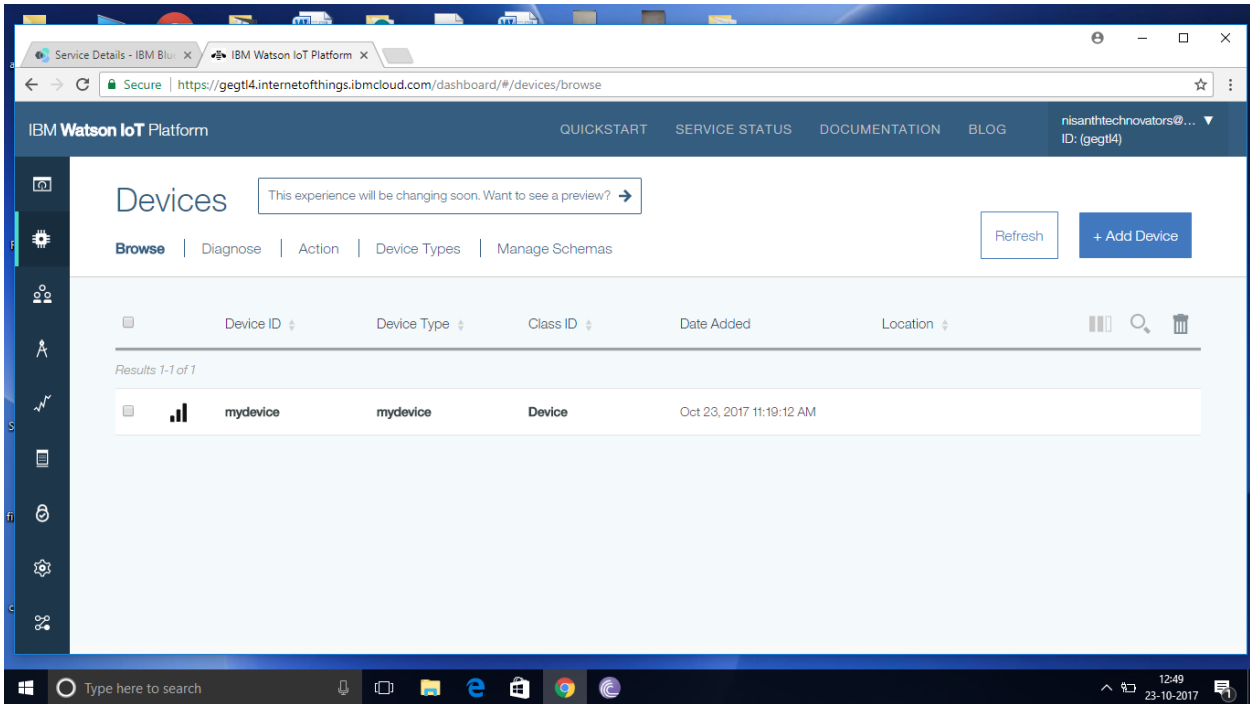
- I have sent DHT-11 Sensors data to ibm bluemix .To get the code u need to login into IOT GYAN.
- Then I get the image as follows in my pi's shell:

A screenshot of a Raspberry Pi desktop environment. The top taskbar shows several open applications: 'Raspberry Pi GPIO Pi...', 'dht11toibmiot.py - /h...', 'credentials.txt', 'pi@raspberrypi: ~', and '*Python 2.7.13 Shell*'. The active window is the Python 2.7.13 Shell, which displays the output of a script. The output shows a successful connection to IBM Watson IoT and several data points for temperature and humidity, along with some 'SensorData Invalid' messages. The status bar at the bottom right indicates 'Ln: 6 Col: 0'.

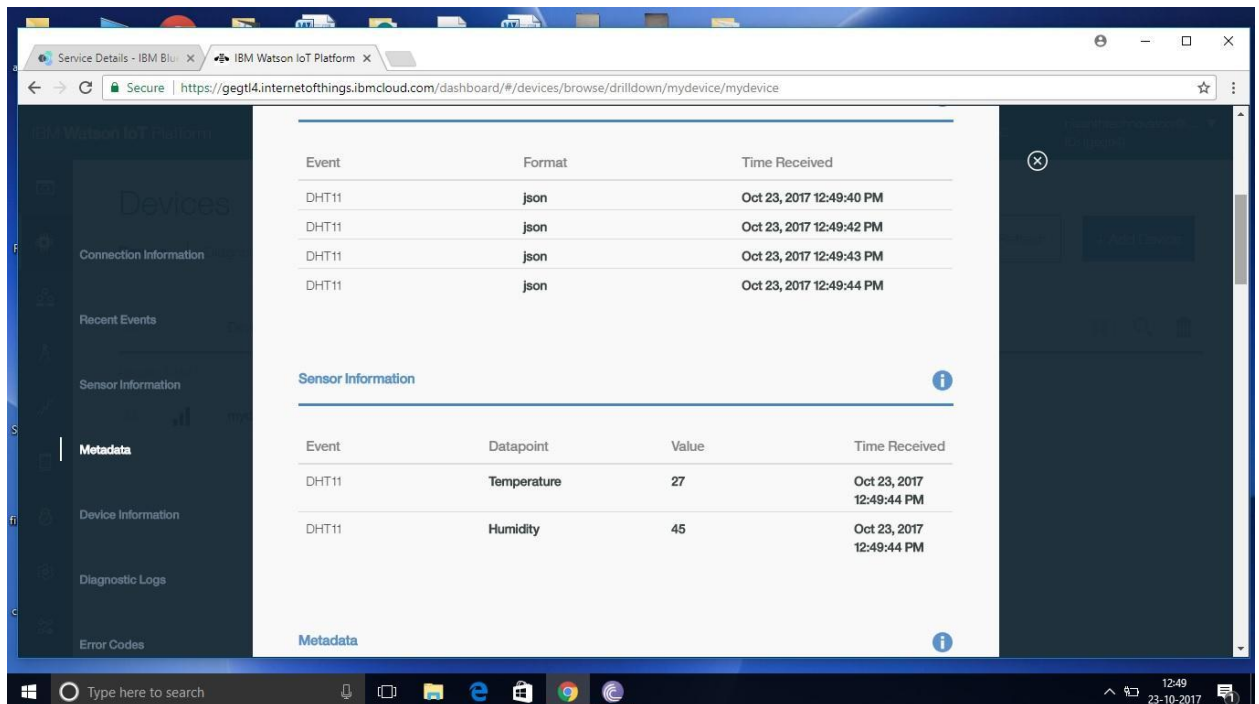
```
Python 2.7.13 (default, Jan 19 2017, 14:48:08)
[GCC 6.3.0 20170124] on linux2
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/pi/Downloads/dht11toibmiot.py =====
2017-10-23 07:10:37.768 ibmiotf.device.Client INFO Connected successfully: d:gegt14:mydevice:mydevice
Published Temperature = 28 C Humidity = 50 % to IBM Watson
SensorData Invalid
Published Temperature = 28 C Humidity = 50 % to IBM Watson
SensorData Invalid
Published Temperature = 28 C Humidity = 50 % to IBM Watson
SensorData Invalid
Published Temperature = 28 C Humidity = 50 % to IBM Watson
Published Temperature = 29 C Humidity = 50 % to IBM Watson
Published Temperature = 29 C Humidity = 50 % to IBM Watson
|
```

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 dta is being received.

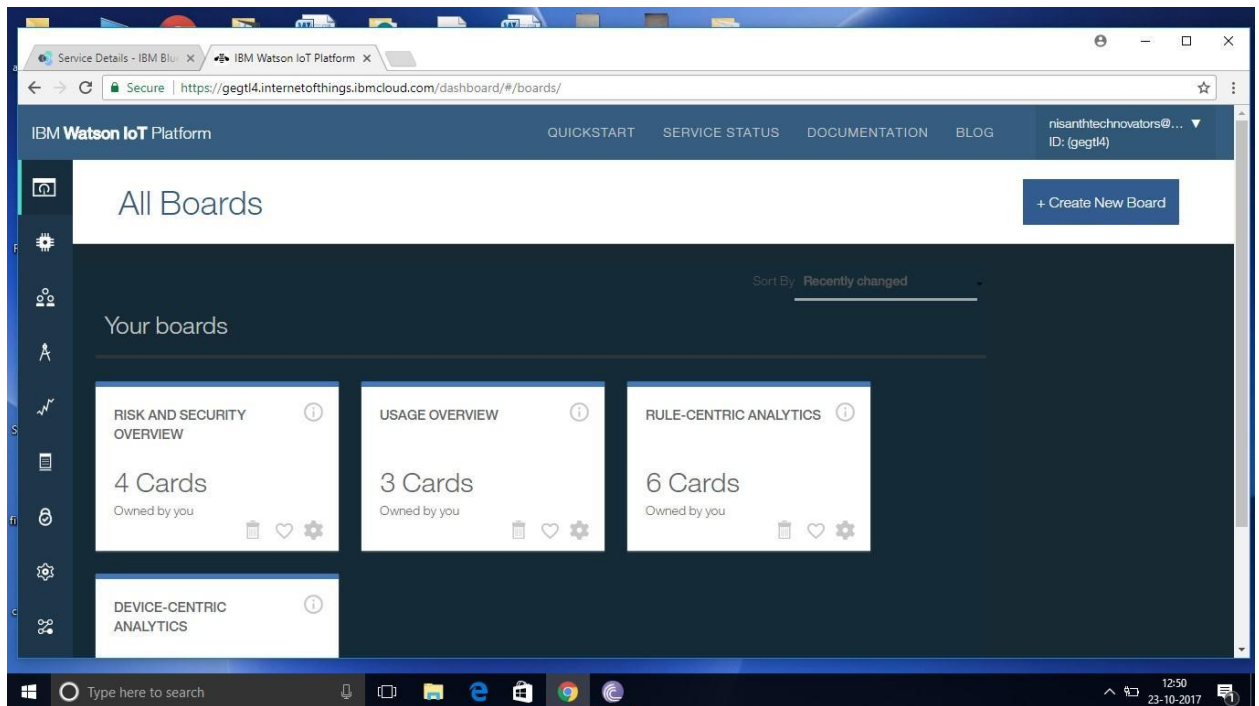


- After double clicking on your created device you can see the received data as shown in image

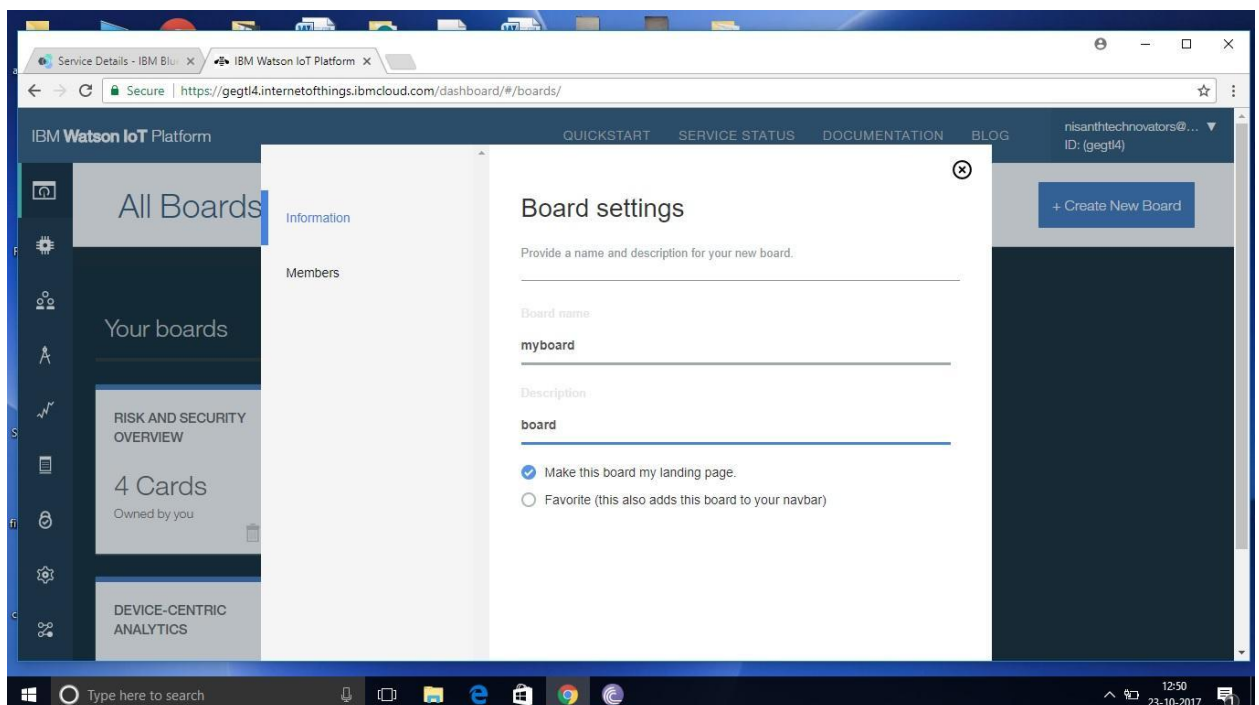


Step-4: Creating boards and cards for visualization of data:

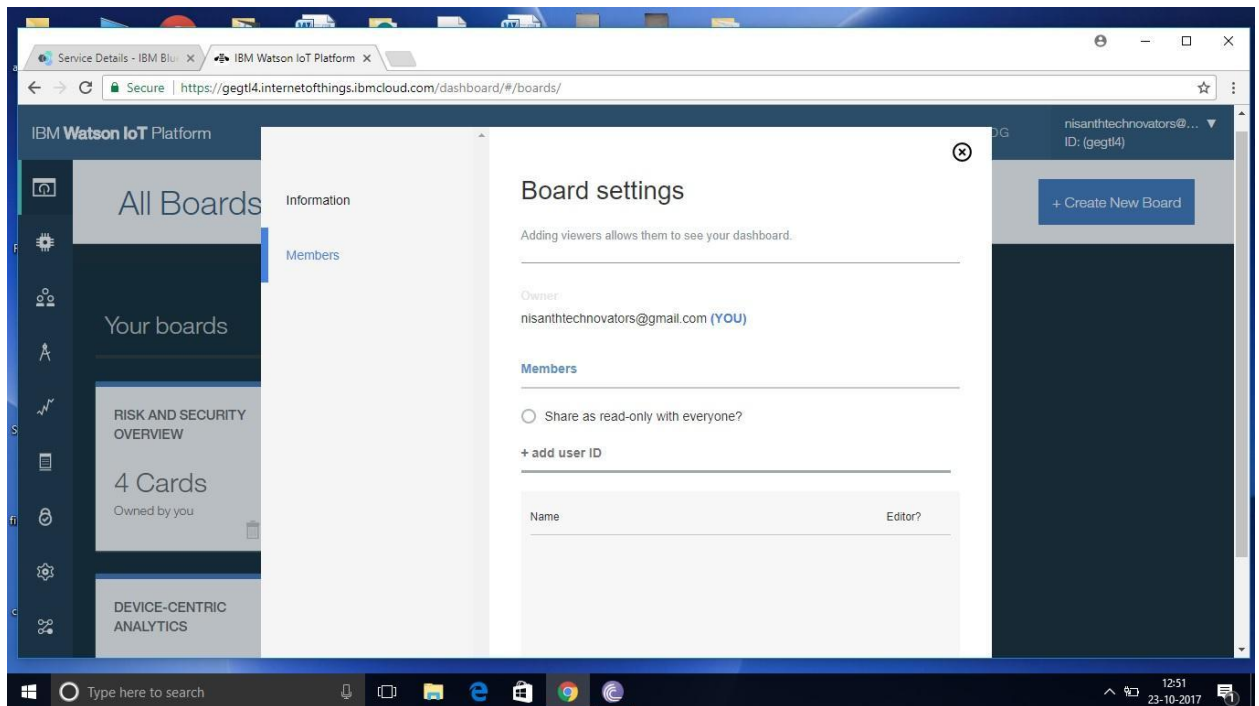
- In your Watson platform you have an option called board .Click on it and you get the following window on your screen



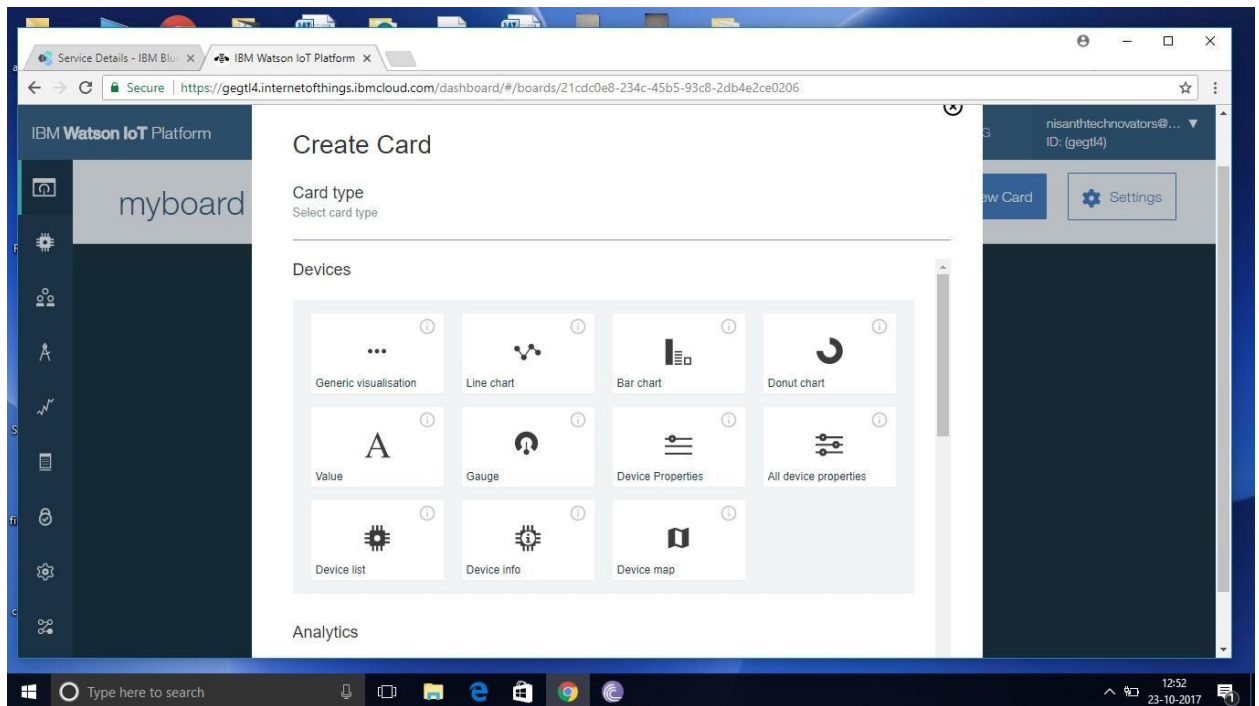
- Click on Create a new board to create a board .
The given below window appears give a name and description to your board as shown in the window below



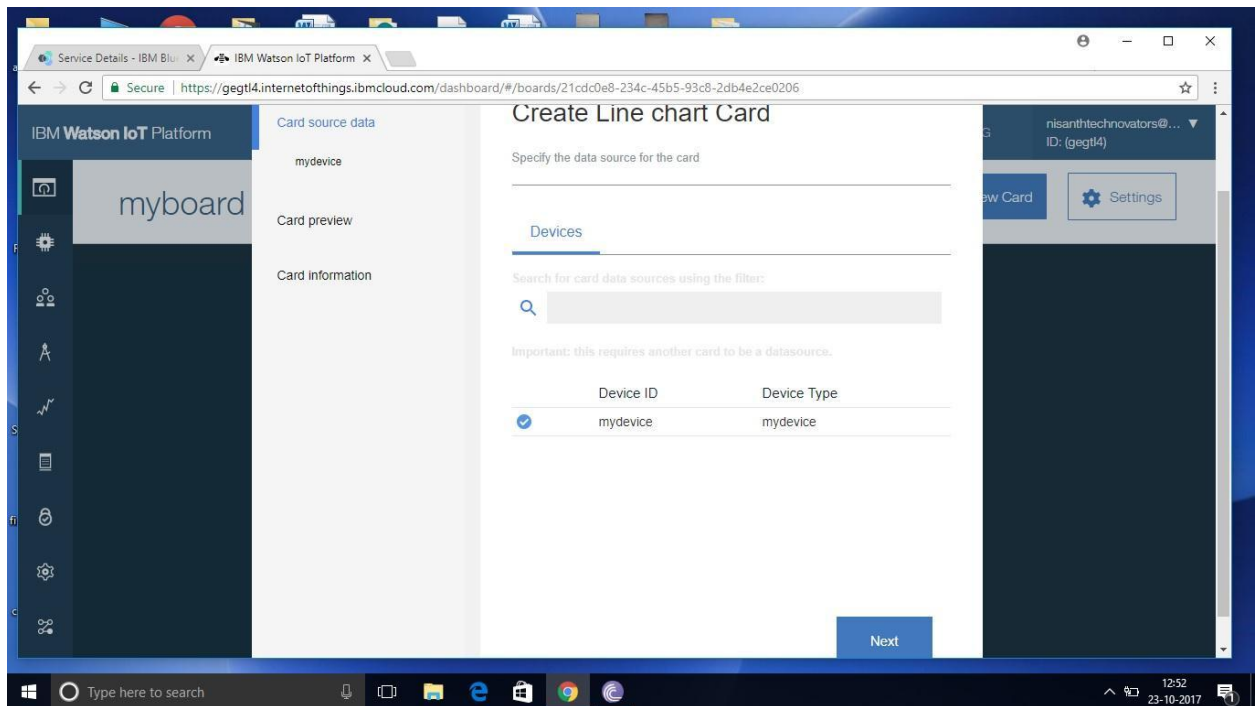
- Then click on Next you get the below window then again click on ADD



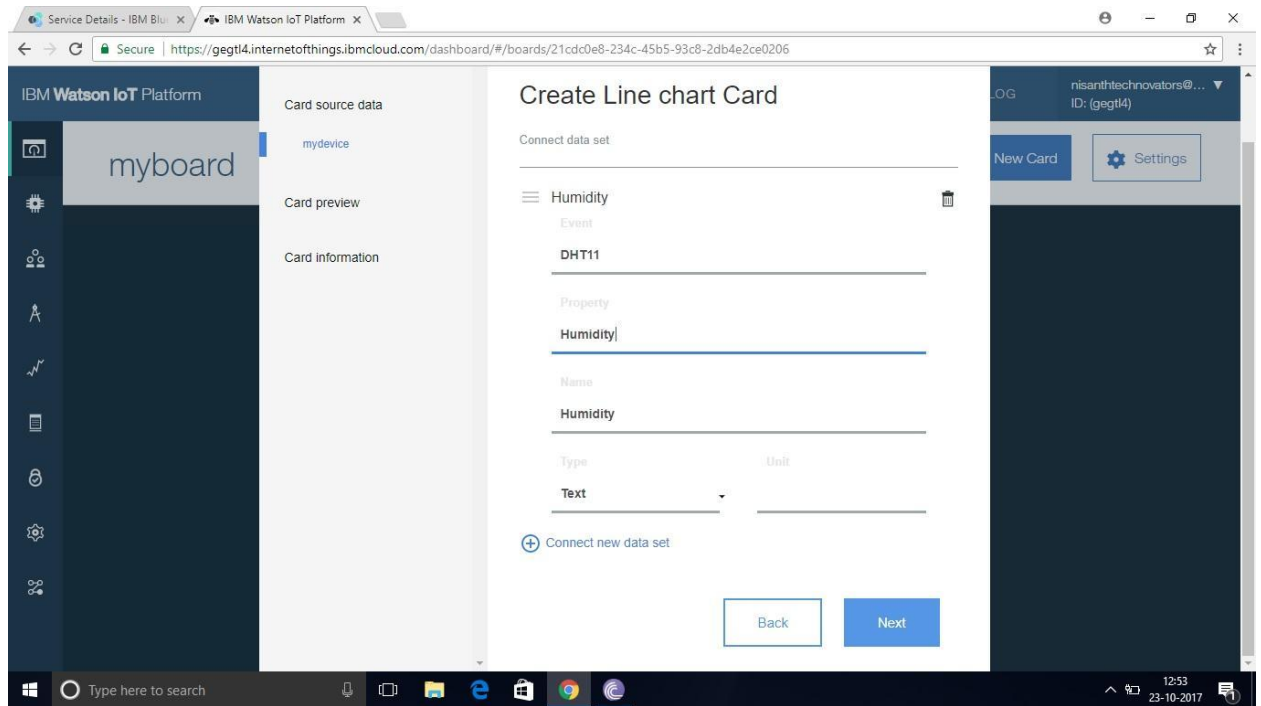
- Then double click on your boards name in the dropdown and click on add new card at the top right corner.



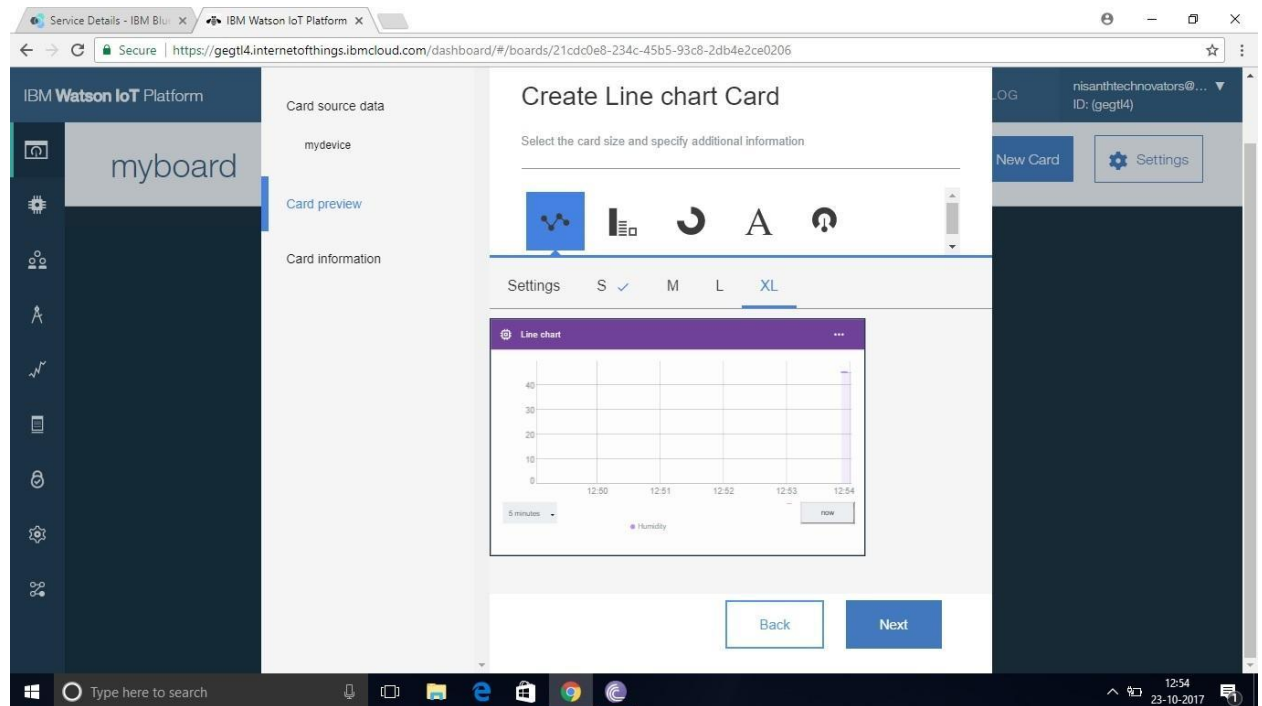
- Select the type of Graph u want accordingly and Click next
- You get the below window

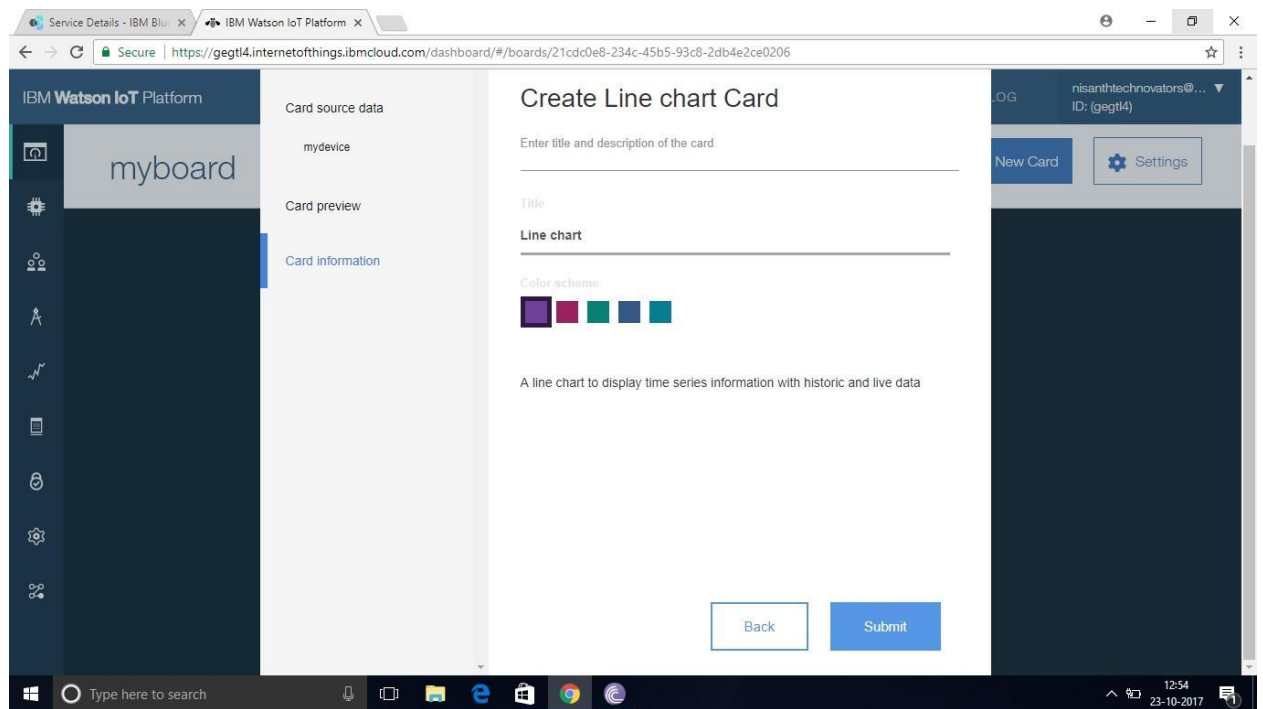


- Tick mark your device name as shown above and then click on next.
- Select the event ,property to be visualized on your graph and click next .In my case it is humidity

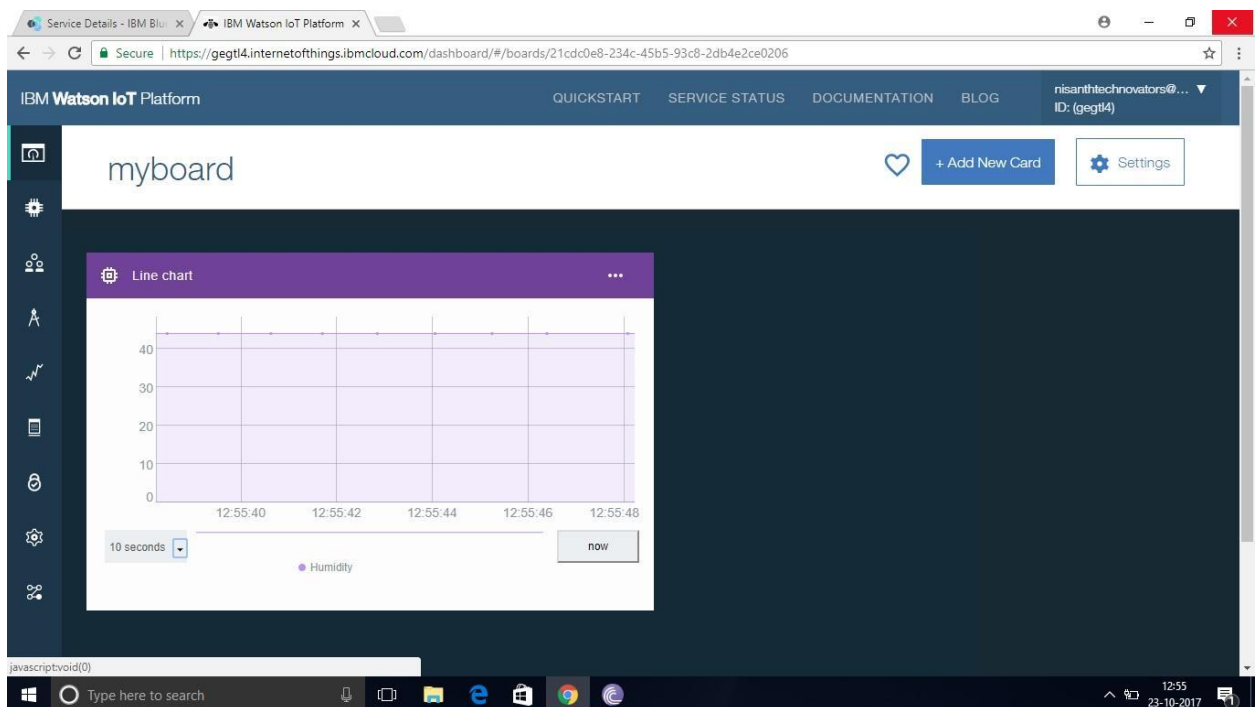


- Then select the size of the graph and color of the graph board you want and click next





- You get your desired data in the form of a graph as shown below



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