

## Sending Data from Raspberry Pi to IBM Watson

<b>Date</b>	19 NOVEMBER 2022
<b>Team ID</b>	PNT2022TMID51029
<b>Project Name</b>	GAS LEAKAGE MONITORING AND ALERTING SYSTEM FOR INDUSTRIES

### 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

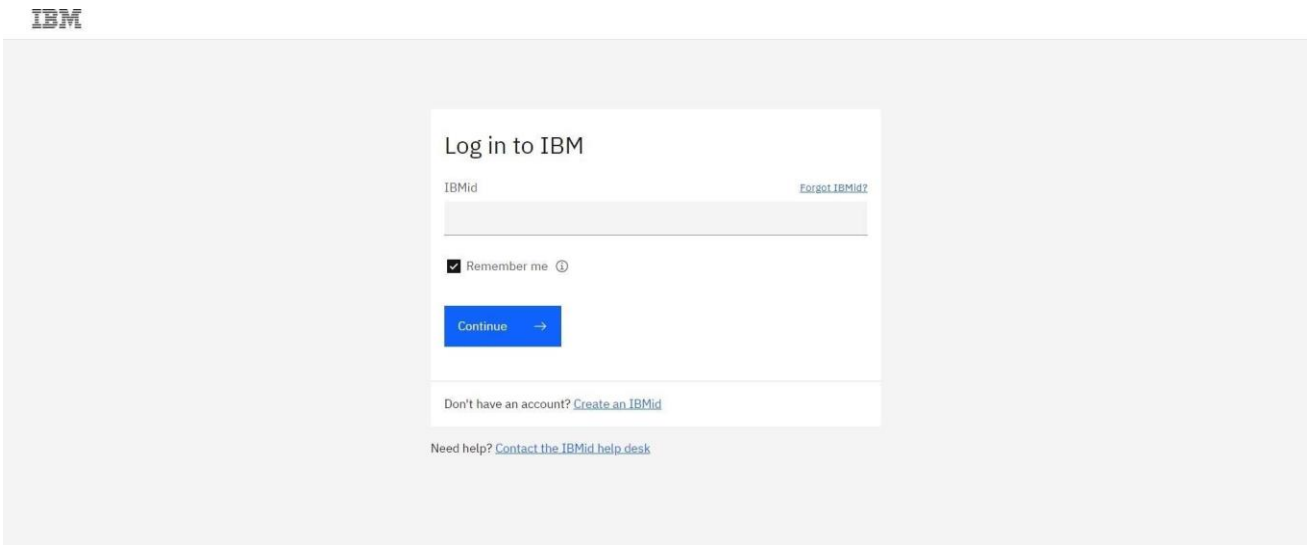
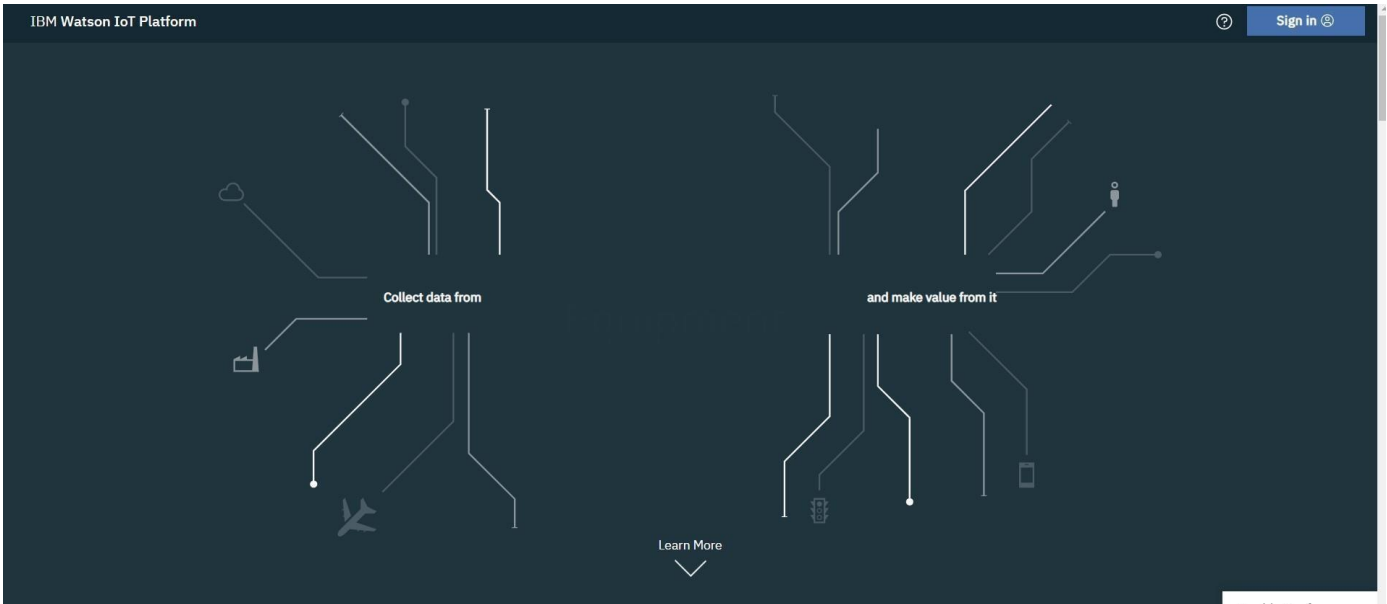
#### SOFTWARE:

- 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.





# Log in to IBM Cloud

[Forgot password?](#)

← 953619104025@smartinternz.com

Password

.....



Log in

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

IBM Cloud

Search resources and products...

Catalog

Manage

LAVANYADEVI K's Acco...

?

Resource list

Create resource +

Name	Group	Location	Product	Status	Tags
Filter by name or IP address...	Filter by group or org...	Filter...	Filter...	Filter...	Filter...
Compute (1+)					
Containers (0)					
Networking (0)					
Storage (0)					
AI / Machine Learning (0)					
Analytics (0)					
Blockchain (0)					
Databases (2+)					
Developer tools (0+)					
Logging and monitoring (0)					
Migration (0)					

Waiting for papi.walkme.com...

Activate Windows  
Go to Settings to activate Windows.

- Check all details and click on create.

The screenshot shows the 'Internet of Things Platform' creation page in the IBM Cloud console. The page is divided into a main content area and a right-hand summary sidebar.

**Main Content Area:**

- Header:** 'Catalog / Internet of Things Platform'. A description states: '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.'
- Tabs:** 'Create' (active) and 'About'.
- Left Sidebar:** Metadata including Type (Service), Provider (IBM), Last updated (08/15/2022), Category (Internet of Things), Compliance (IAM-enabled), and Location (Frankfurt).
- Form:**
  - Select a location:** A dropdown menu showing 'Frankfurt (eu-de)'.
  - Select a pricing plan:** A note states 'Displayed prices do not include tax. Monthly prices shown are for country or location: [United States](#)'.
  - Table:** A table with columns 'Plan', 'Features', and 'Pricing'. The 'Lite' plan is selected, showing features like 'Includes up to 500 registered devices, and a maximum of 200 MB of each data metric' and 'Maximum of 500 registered devices'.

**Right Sidebar (Summary):**

- Header:** 'Internet of Things Platform' with a 'Free' tag.
- Details:** Location: Frankfurt, Plan: Lite, Service name: Internet of Things Platform-0g, Resource group: Default.
- Agreement:** A checkbox 'I have read and agree to the following license agreements:' is checked, with a 'Terms' link.
- Buttons:** 'Create' (blue) and 'Add to estimate' (grey).

- click on Launch

The screenshot shows the 'Internet of Things Platform-0g' dashboard. The interface includes a left navigation menu, a main content area with a large illustration and text, and a bottom section for the 'IBM Watson IoT Platform Journey'.

**Navigation and Header:**

- Resource list /** 'Internet of Things Platform-0g' with status 'Active' and 'Add tags' link.
- Actions:** 'Details' and 'Actions...' dropdown.
- Left Menu:** 'Manage' (selected), 'Plan', 'Connections'.

**Main Content Area:**

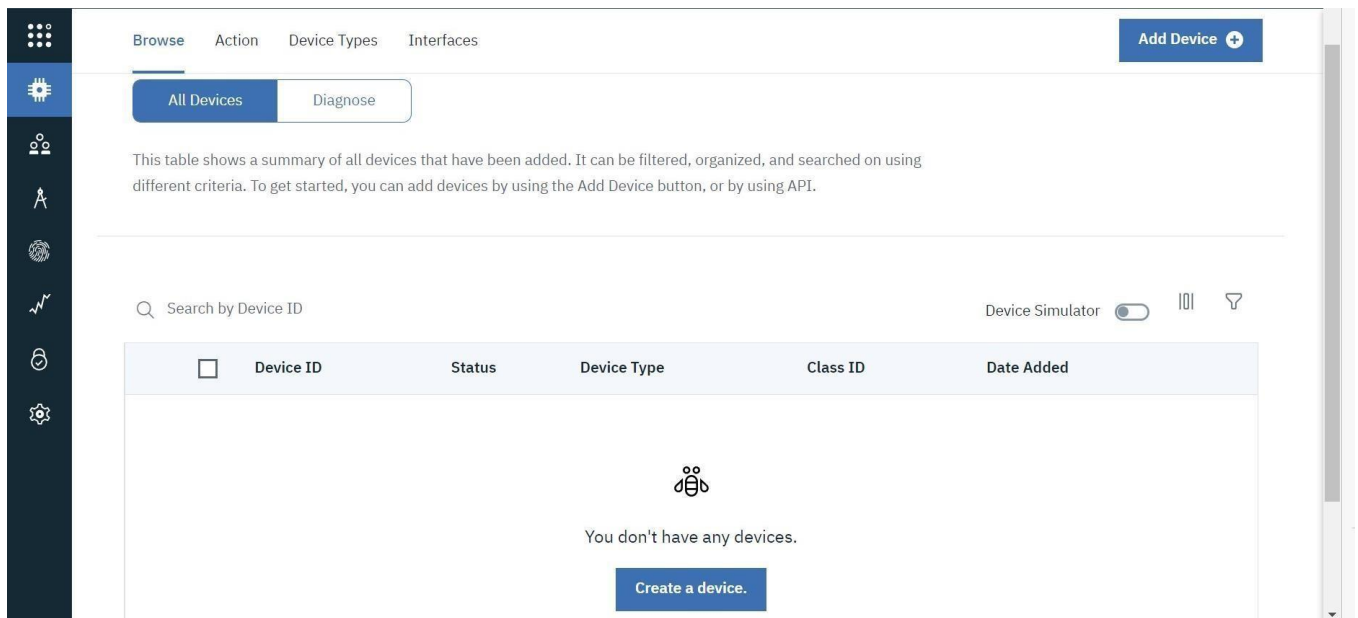
- Illustration:** A central graphic showing a device connected to a network of nodes.
- Text:** 'Let's get started with IBM Watson IoT Platform'. Subtext: 'Securely connect, control, and manage devices. Quickly build IoT applications that analyze data from the physical world.'
- Buttons:** 'Launch' (blue) and 'Docs' (grey).

**Bottom Section:**

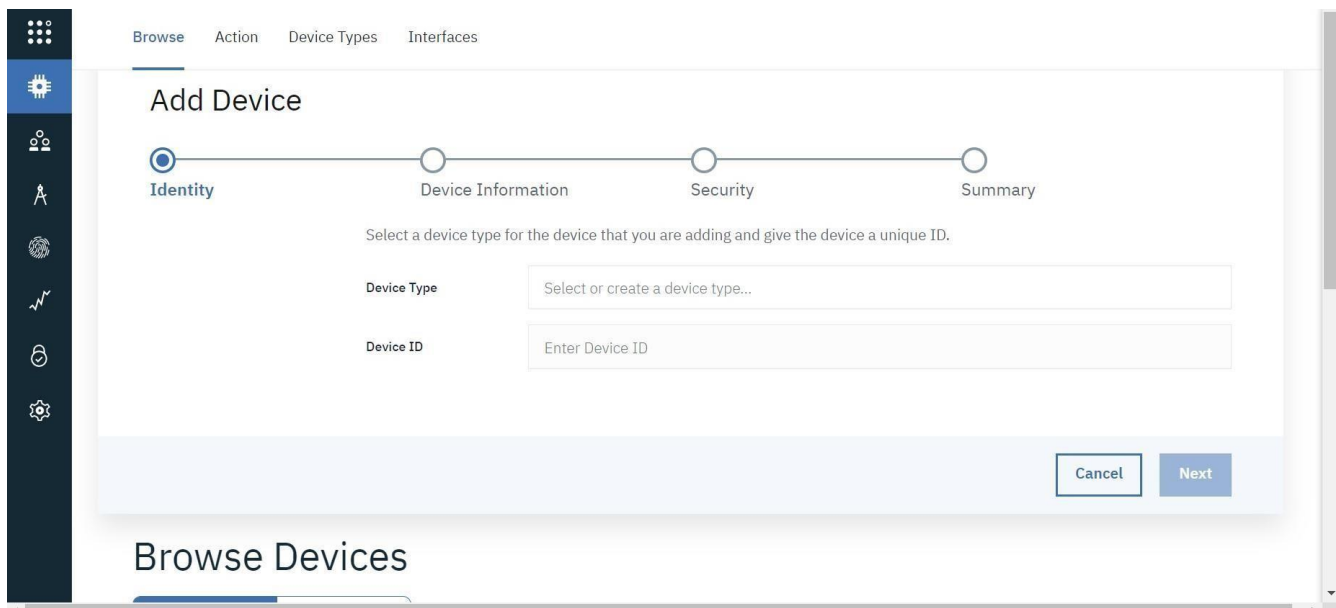
- Header:** 'Ready for the next level?'.
- Journey:** 'IBM Watson IoT Platform Journey'.
- Progress:** A progress bar with two steps: 'Lite' (checked) and 'Non-Production' (unchecked).

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.

○ Click on Finish

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

Edit Metadata

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

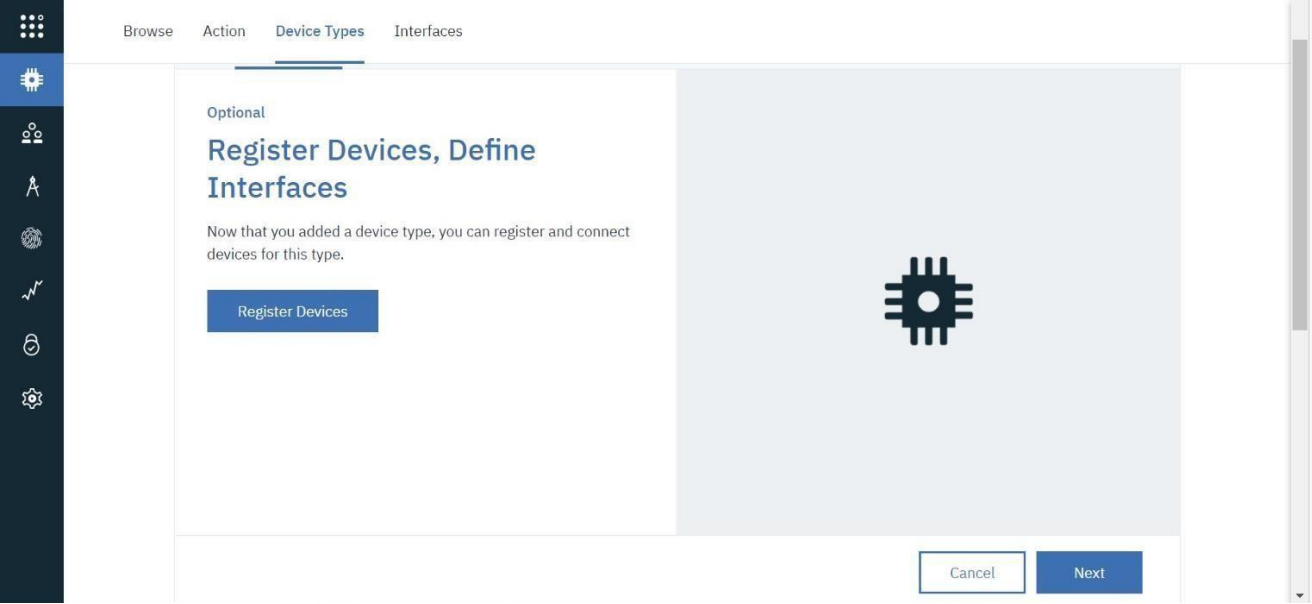
Descriptive Location

Enter Descriptive Location

Back

Finish

Register Device.



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



## Click on Next

The screenshot shows the 'Add Device' form with the 'Device Information' step selected. The form includes a progress bar at the top with four steps: Identity, Device Information, Security, and Summary. Below the progress bar, a message states: 'You can modify the default device information and enter more information about the device for identification purposes.' The form contains two columns of input fields. The left column has fields for 'Serial Number', 'Model', 'Description', and 'Hardware Version'. The right column has fields for 'Manufacturer', 'Device Class', 'Firmware Version', and 'Descriptive Location'. Each field has a placeholder text 'Enter [field name]'. At the bottom left of the form is a button labeled 'Add Metadata' with a plus icon.

Browse Action Device Types Interfaces

### Add Device

Identity Device Information Security Summary

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 +

## Click on Next

The screenshot shows the 'Add Device' form with the 'Security' step selected. The progress bar at the top shows 'Identity' and 'Device Information' as completed steps, 'Security' as the current step, and 'Summary' as the next step. The main content area is titled 'There are two options for selecting a device authentication token.' and is split into two columns. The left column is titled 'Auto-generated authentication token (default)' and contains text explaining that the service will generate an 18-character alphanumeric token. The right column is titled 'Self-provided authentication token' and contains text explaining that the user must provide their own token between 8 and 36 characters. Below this, there is an 'Authentication Token' label and a text input field with the placeholder 'Enter an optional token' and an information icon. At the bottom, a note states: 'Make a note of the generated token. Lost authentication tokens cannot be recovered. Tokens are encrypted before being stored.' and a sub-note says: 'Authentication token are encrypted before we store them.'

Browse Action Device Types Interfaces

Identity Device Information Security Summary

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 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 token are encrypted before we store them.

Finish

○ Device is created

IBM Watson IoT Platform

953619104025@smartinternz.com  
ID: a49saj

Browse

Action

Device Types

Interfaces

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.

Q Search by Device ID

Device Simulator ☒

<input type="checkbox"/>	Device ID	Status	Device Type	Class ID	Date Added
> <input type="checkbox"/>	weather_device_1	Disconnected	weather_device	Device	14 Nov 2022 10:41

Items per page 50 | 1-1 of 1 item

1 of 1 page

<

1

>

## 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
pi@raspberrypi:~$ curl -LO https://github.com/ibm-messaging/iot-raspberrypi/releases/download/1.0.2.1/iot_1.0-2_armhf.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 164    0 164    0  0 157    0 --:--:--  0:00:01 --:--:-- 157
100 609    0 609    0  0 457    0 --:--:--  0:00:01 --:--:-- 457
100 110k 100 110k  0  0 291k    0 0:00:03  0:00:03 --:--:-- 48190
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:armhf (1.0.1t-1+deb8u6) ...
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/releases/download/1.0.2.1/iot_1.0-2_armhf.deb
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 164    0 164    0  0 157    0 --:--:--  0:00:01 --:--:-- 157
100 609    0 609    0  0 457    0 --:--:--  0:00:01 --:--:-- 457
100 110k 100 110k  0  0 291k    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-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.
```

- 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 (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-2.10.4-py2.py3-none-any.whl (88kB)
    100% |#####| 92kB 1.0MB/s
Collecting requests-toolbelt>=0.7.0 (from ibmiotf)
  Downloading requests-toolbelt-0.8.0-py2.py3-none-any.whl (54kB)
    100% |#####| 61kB 1.0MB/s
Collecting xmltodict>=0.10.2 (from ibmiotf)
  Downloading xmltodict-0.11.0-py2.py3-none-any.whl
Collecting urllib3<4.23.0, >=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-2.0-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/f9/45/bbc33ad957e82f7b71ba80e316d65a83d9d735a6d12e0c0418
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/acdc8f289011b7be7de71deebef0642f8b3be0313dfff0493
Successfully built ibmiotf dicttoxml paho-mqtt
Installing collected packages: dicttoxml, iso8601, paho-mqtt, pytz, urllib3, idna, chardet, certifi, requests, requests-toolbelt, xmltodict, ibmiotf
Successfully installed certifi-2017.7.27.1 chardet-3.0.4 dicttoxml-1.7.4 ibmiotf-0.3.0 idna-2.0 iso8601-0.1.12 paho-mqtt-1.3.1 pytz-2017.2 requests-2.10.4 requests-toolbelt-0.8.0 urllib3-1.22 xmltodict-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:

```
File Edit Shell Debug Options Window Help
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/dhtlltoibmiot.py =====
2017-10-23 07:10:37,706 ibmiot.device.Client INFO Connected successfully: d:gegl14: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 data is being received.

IBM Watson IoT Platform

953619104025@smartinternz.com  
ID: a49saj

BrowseActionDevice TypesInterfacesAdd Device +

# Browse Devices

All DevicesDiagnose

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.

Q Search by Device ID

Device Simulator ☒

|||

Y

<input type="checkbox"/>	Device ID	Status	Device Type	Class ID	Date Added
> <input type="checkbox"/>	weather_device_1	Disconnected	weather_device	Device	14 Nov 2022 10:41

Items per page 50 | 1–1 of 1 item1 of 1 page<1>

1 Simulation running

Activate Windows  
Go to Settings to activate Windows.

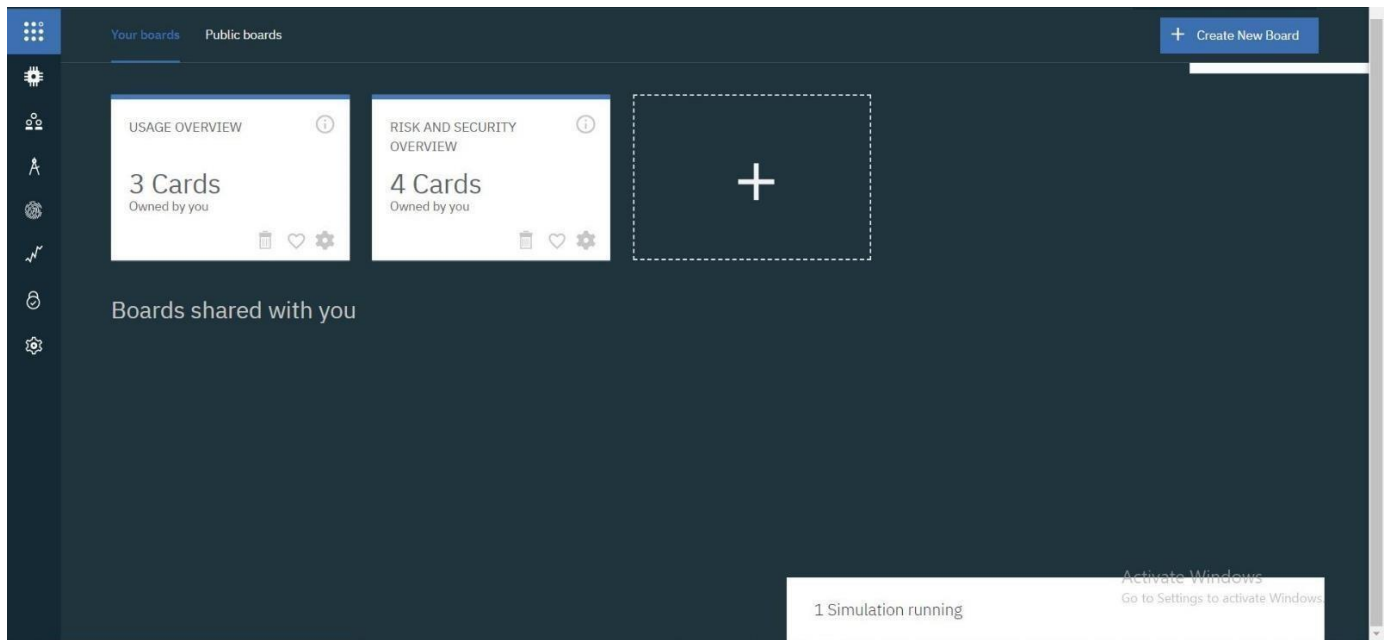
- After double clicking on your created device you can see the received data as shown in image

## 1 Simulation running

Activate Windows  
Go to Settings to activate Windows.

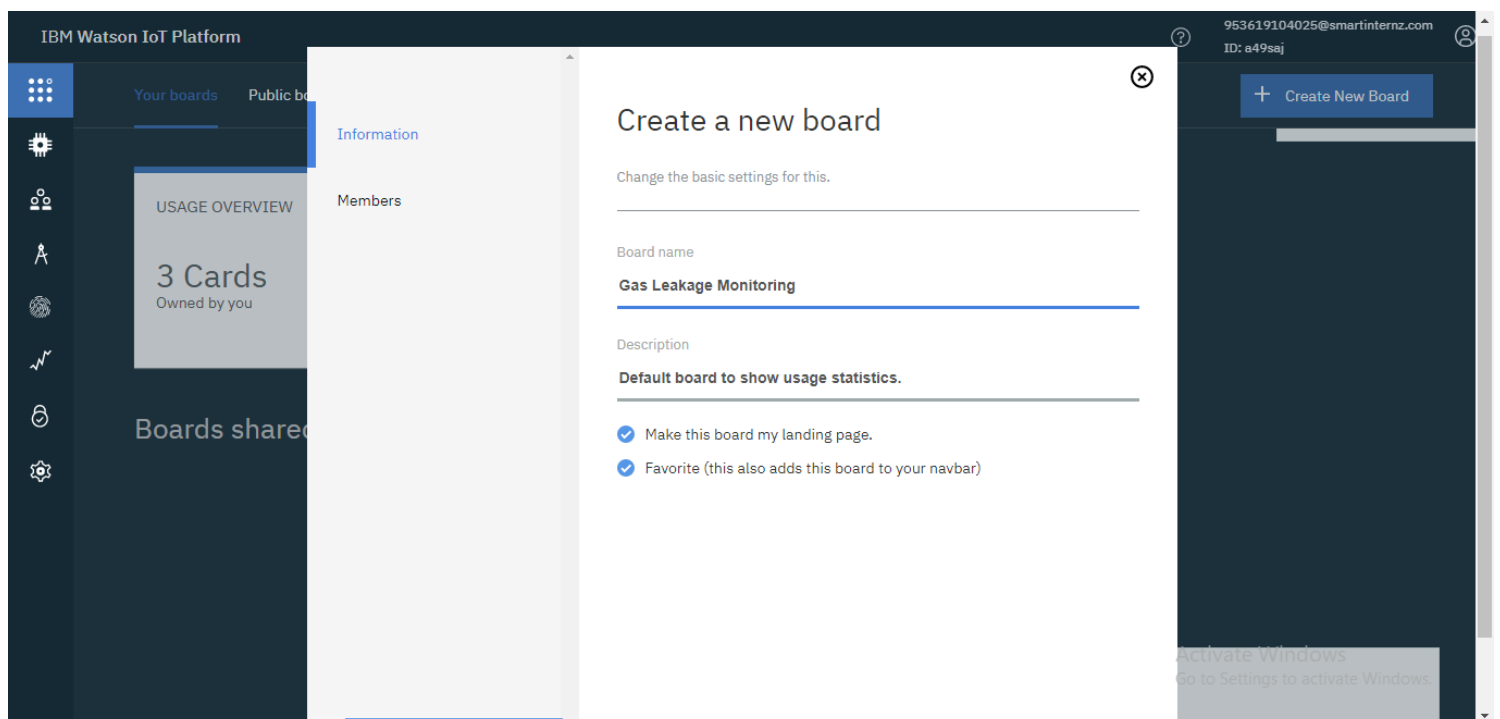
#### Step-4: Creating boards and cards for visualization of data:

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

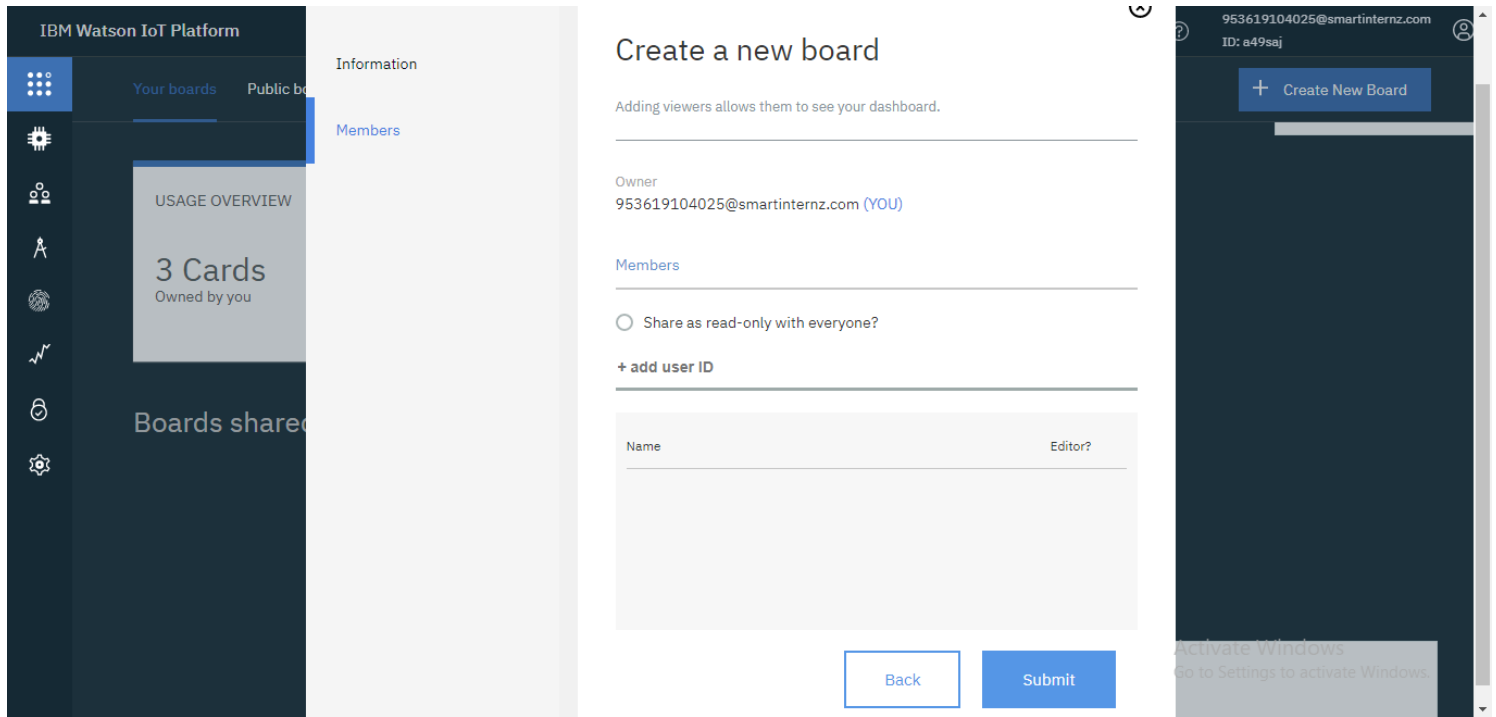


- Click on Create a new board to create a board .

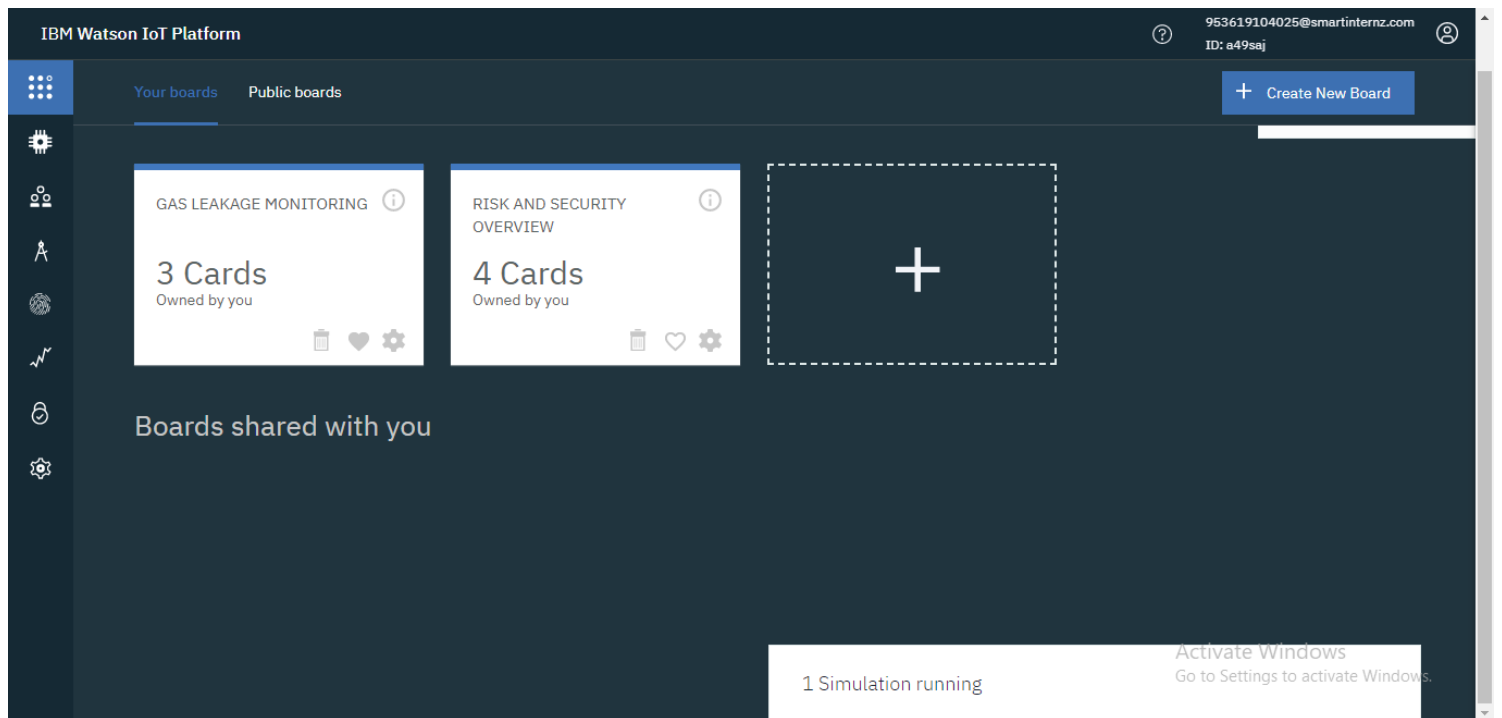
The given below window appears give a name and description to your board as shown in the window below.



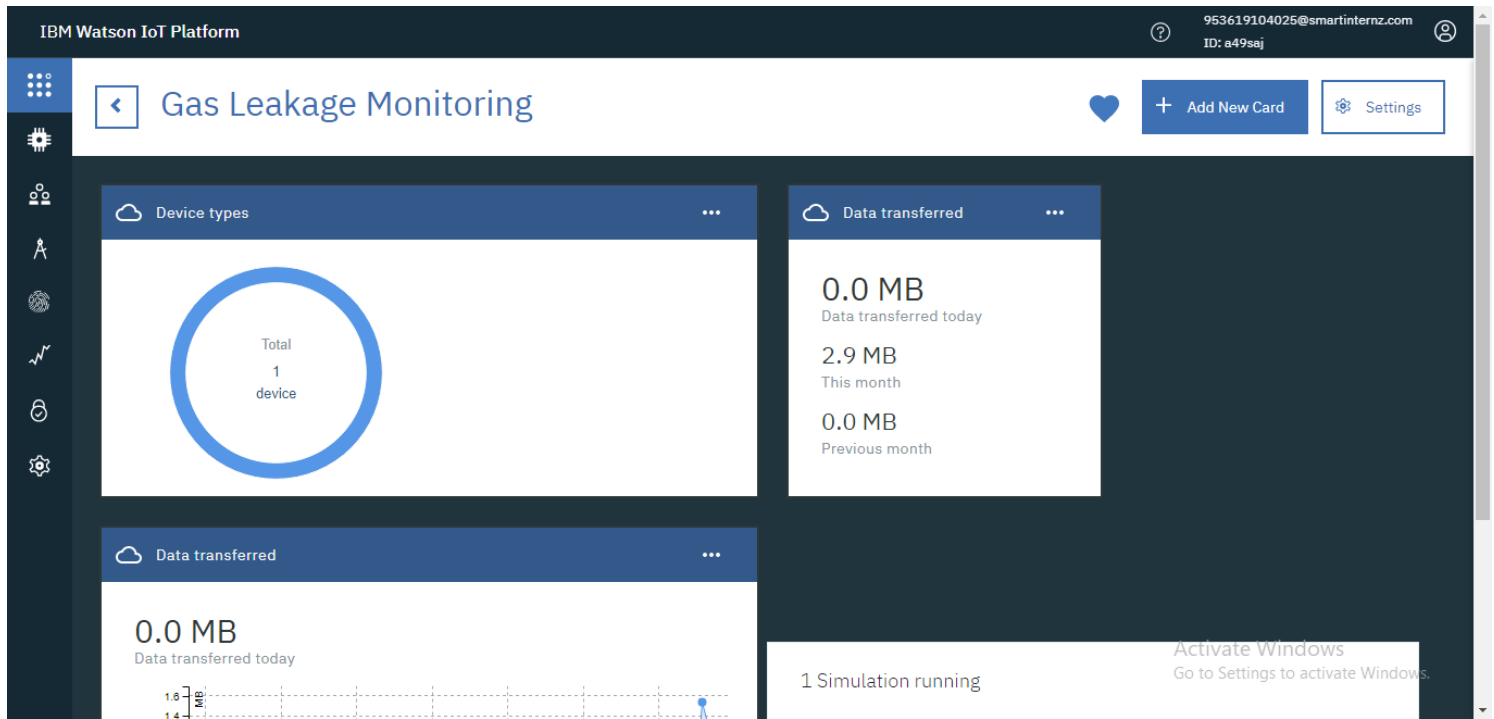
- Then click on Next you get the below window then again click on Submit



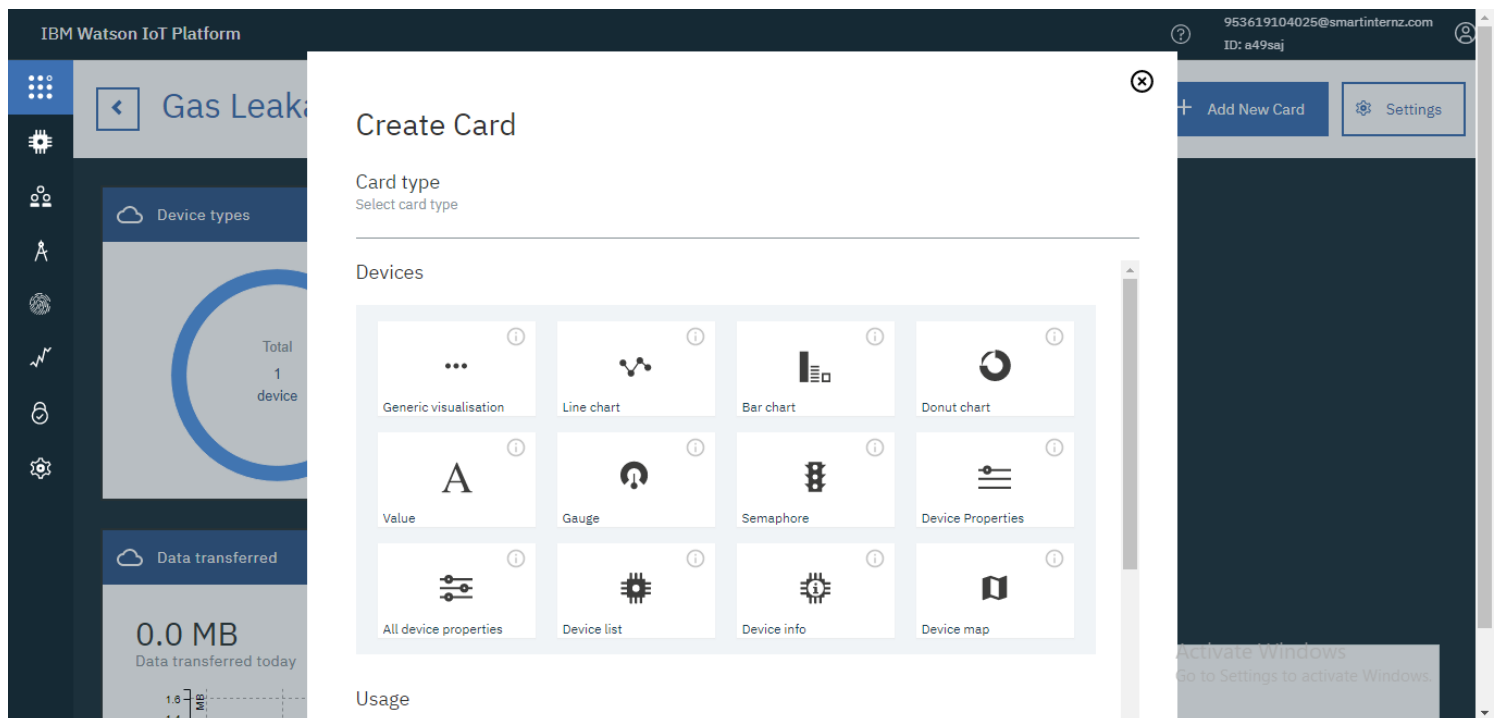
- Then double click on your board name which you have created.



○ Click on Add New Card

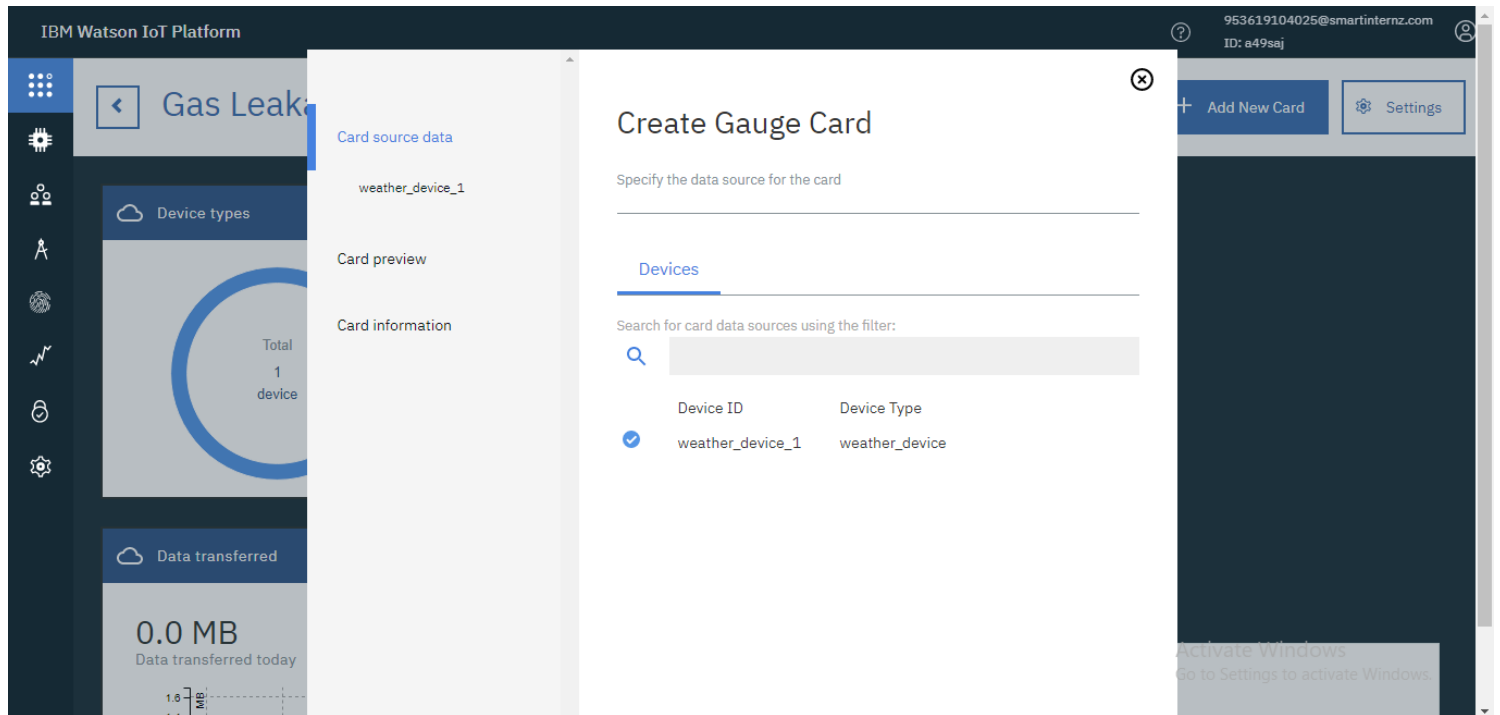


○ Select the type of Graph you want accordingly and click next

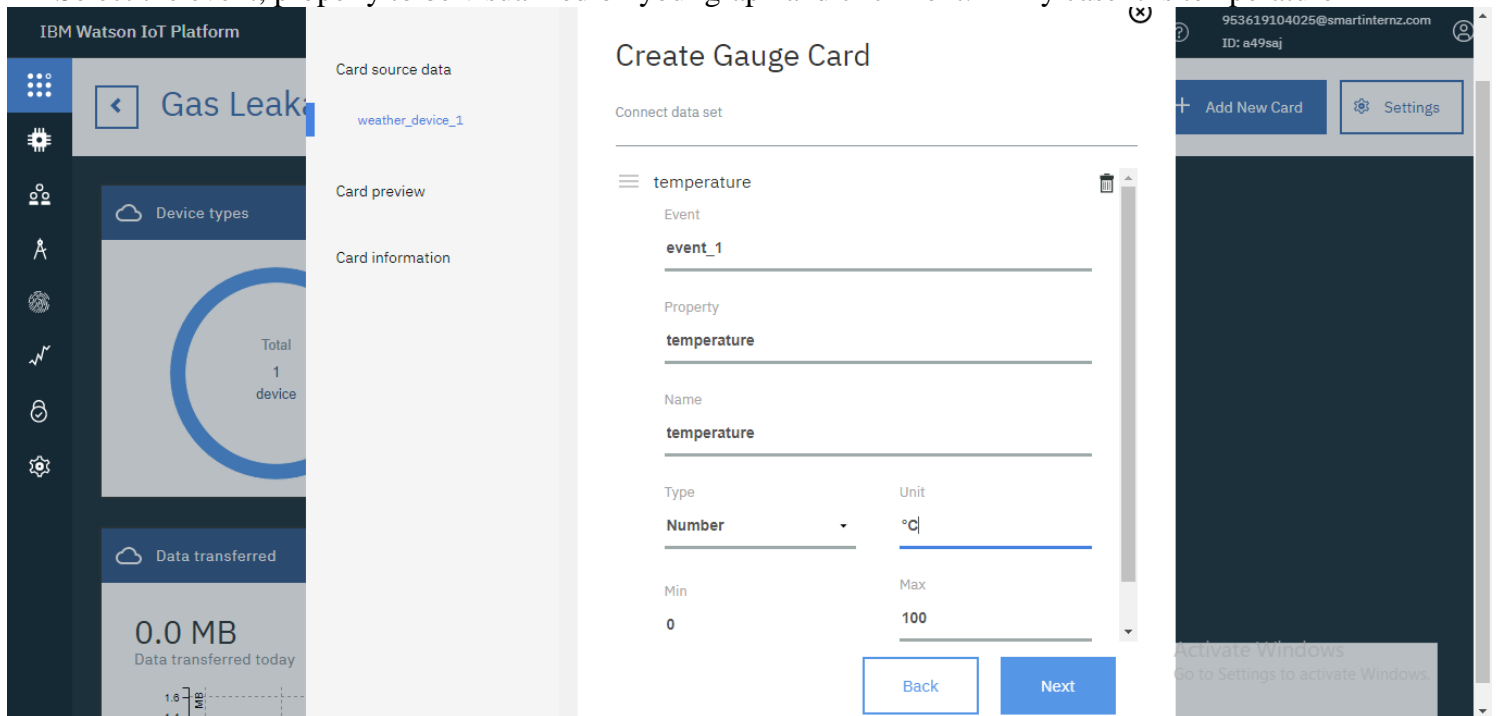




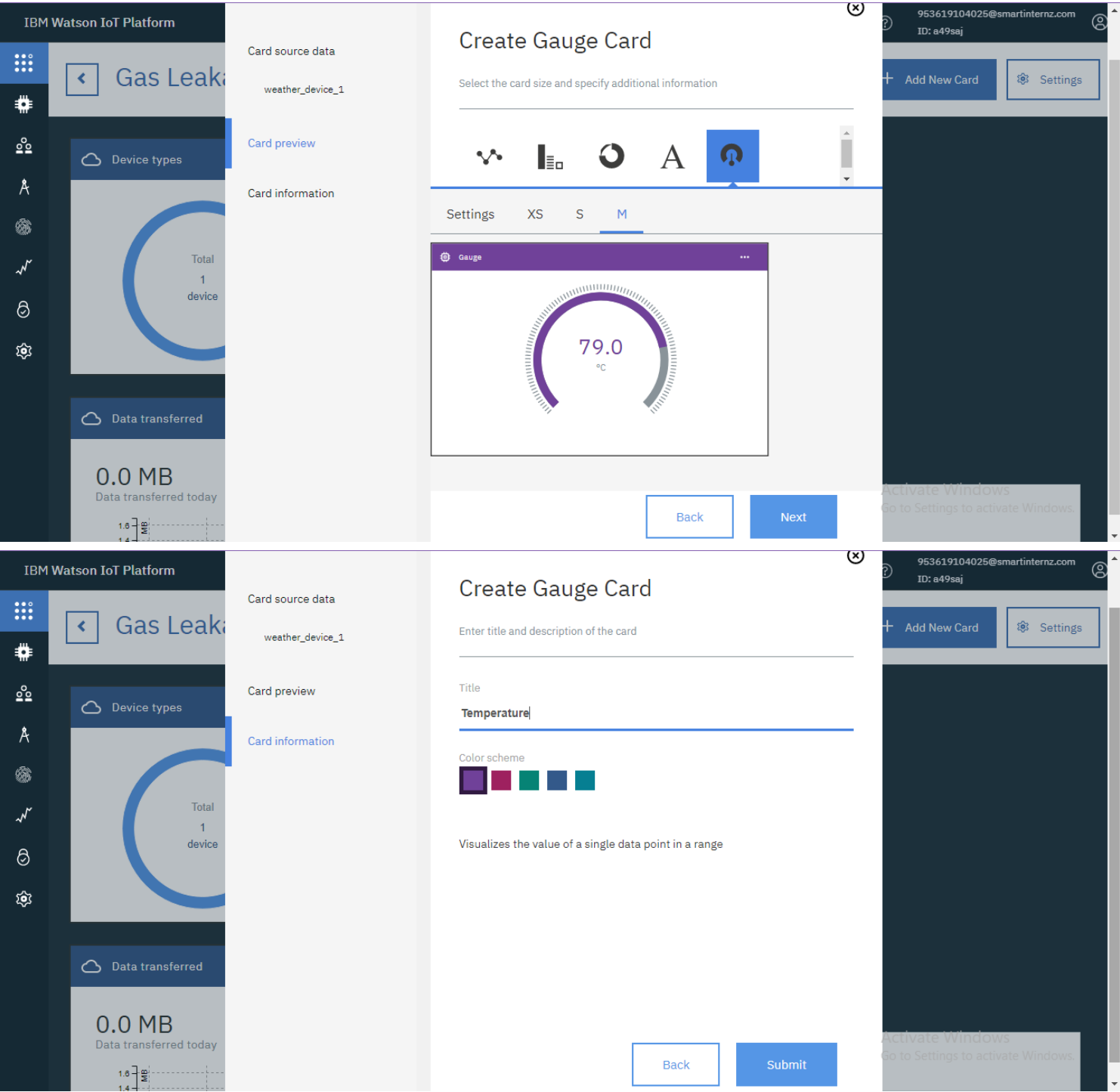
- You get the below window, choose the Device and click on Next.



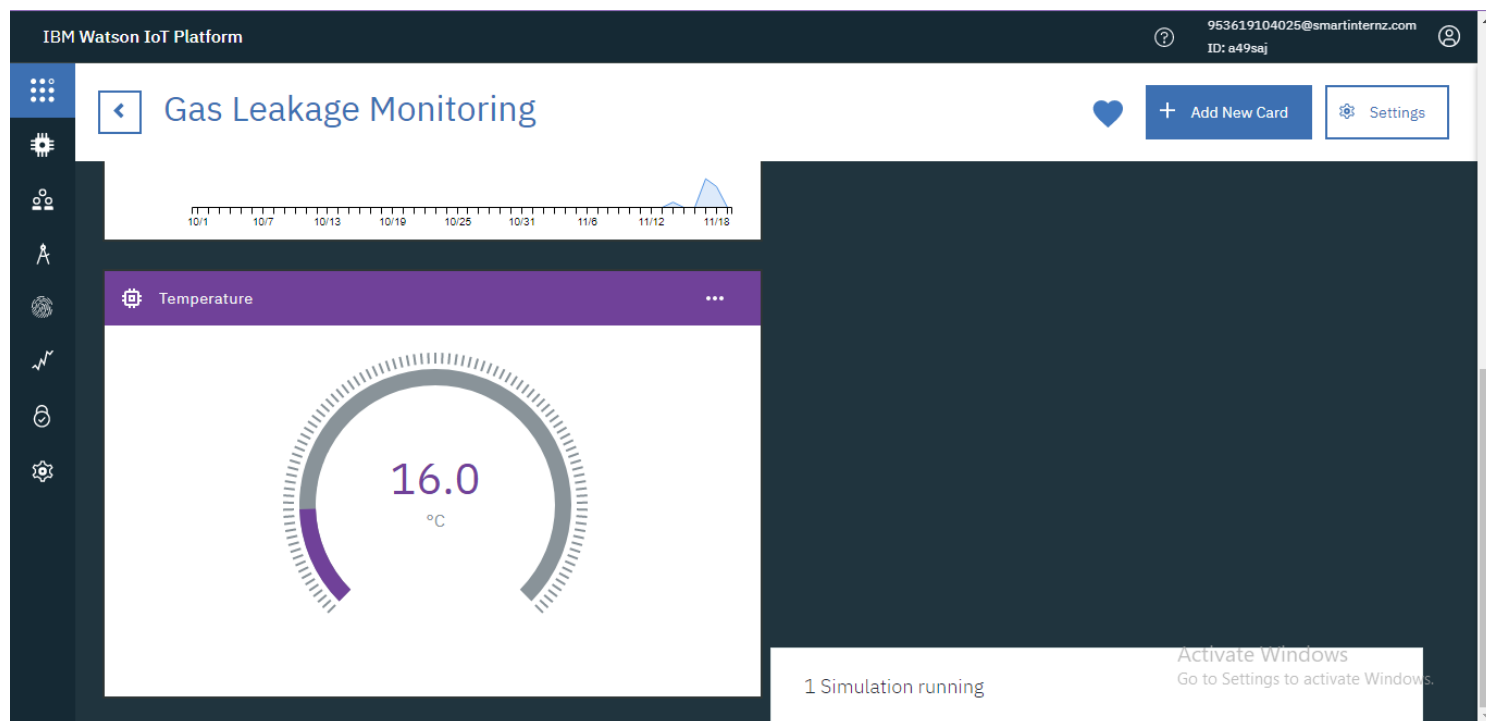
- Select the event, properly to be visualized on your graph and click next. In my case it is temperature



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

