

SENDING DATA FROM RASPBERRY-PI TO IBM WATSON

Date	3 NOVEMBER 2022
Team ID	PNT2022TMID06950
Project Name	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

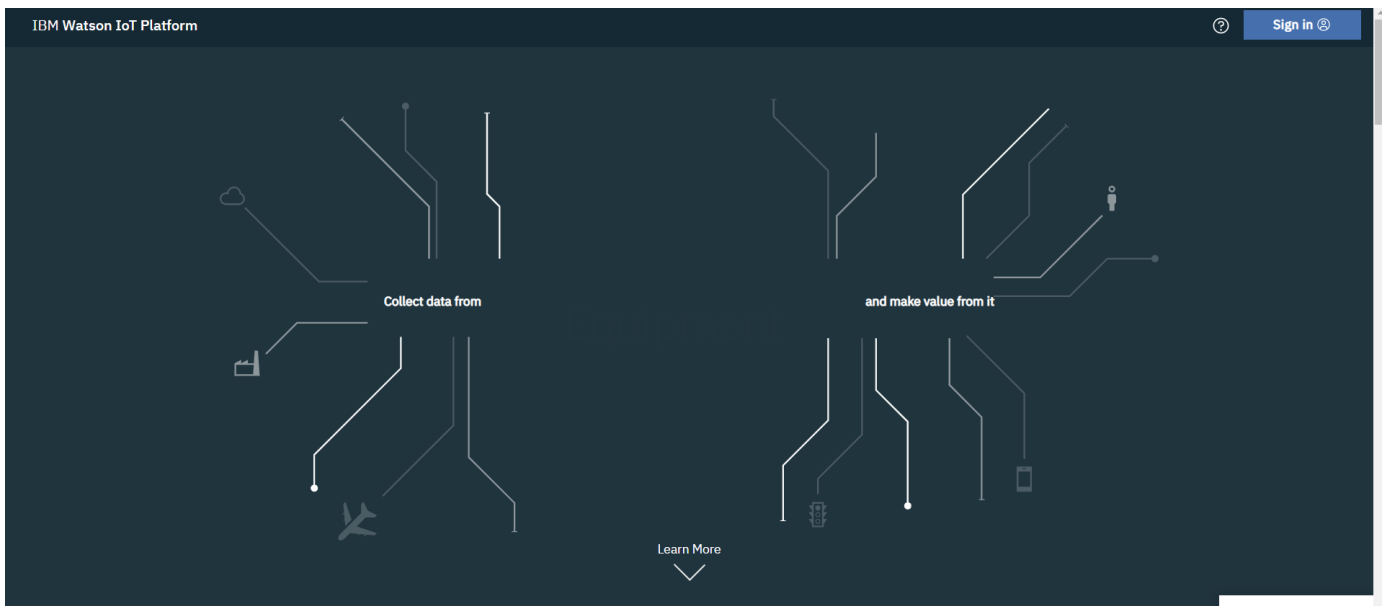
SOFTWARE:

- 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.



IBM

Log in to IBM

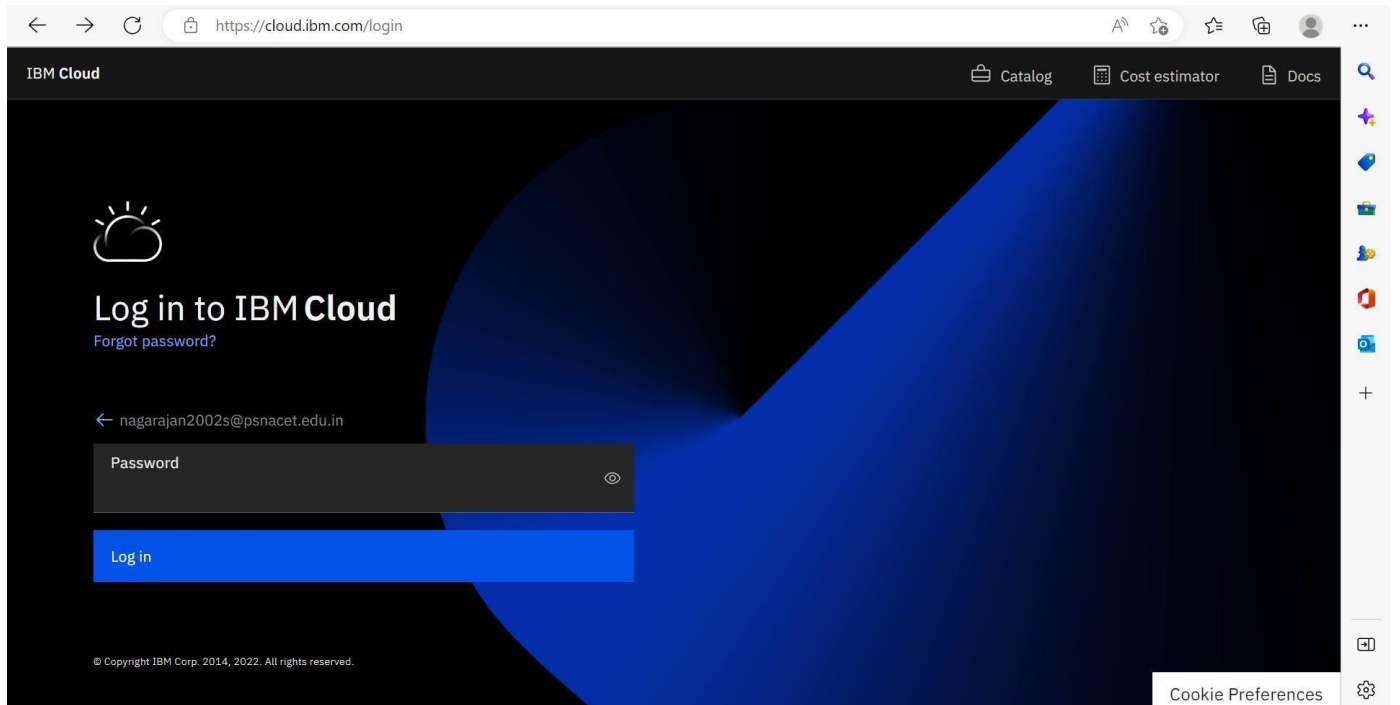
IBMid [Forgot IBMid?](#)

☒ Remember me ⓘ

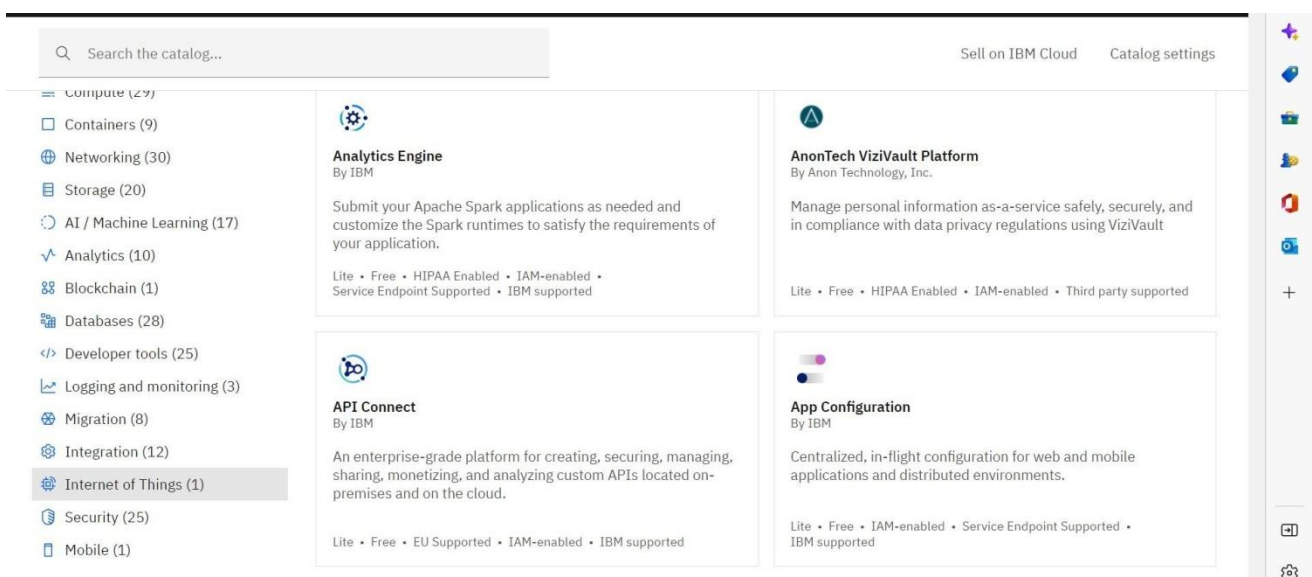
[Continue](#) →

Don't have an account? [Create an IBMid](#)

Need help? [Contact the IBMid help desk](#)




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



➤ Check all details and click on create.

➤ click on Launch

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.

Create

About

Type
Service

Provider
IBM

Last updated
08/15/2022

Category
Internet of Things

Compliance
IAM-enabled

Location
Frankfurt

Select a location

Frankfurt (eu-de)

Select a pricing plan

Displayed prices do not include tax. Monthly prices shown are for country or location: [United States](#)

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

Summary

Internet of Things Platform **Free**

Location: Frankfurt

Plan: Lite

Service name: Internet of Things Platform-0g

Resource group: Default

☒ I have read and agree to the following license agreements:
[Terms](#)

Create

Add to estimate

Resource list /

Internet of Things Platform-0g Active [Add tags](#)

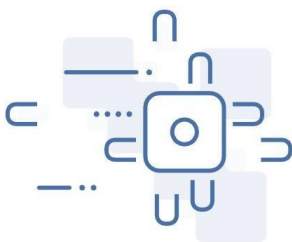
Details

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

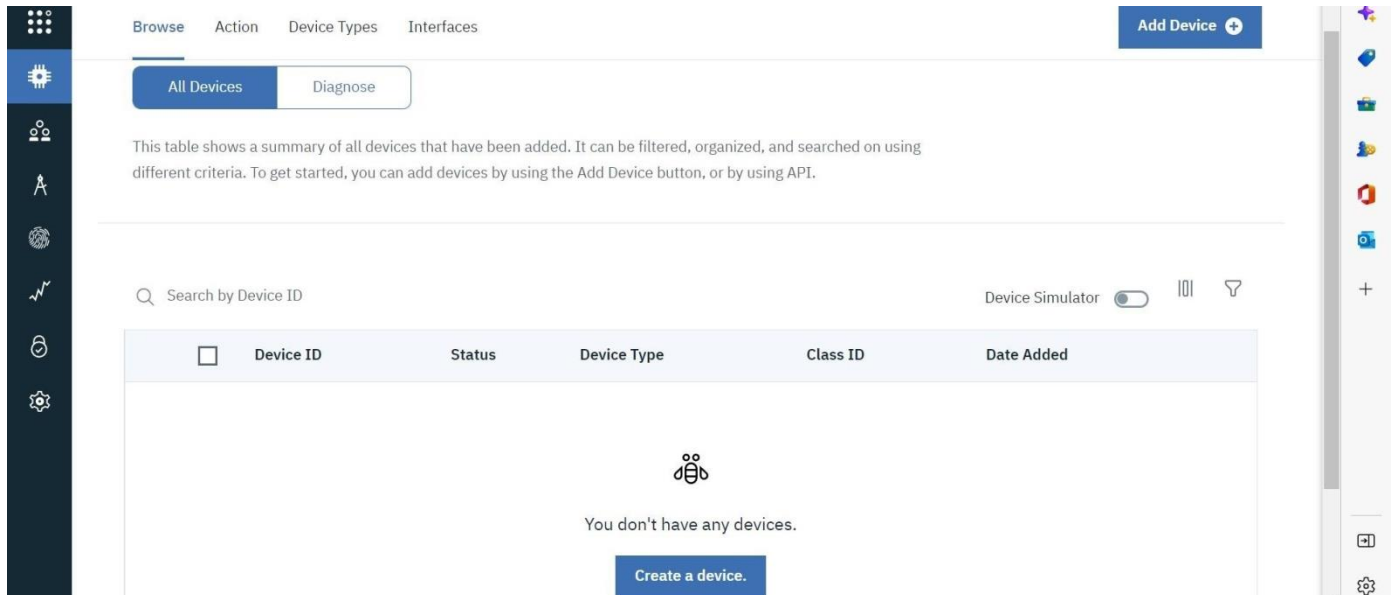
Ready for the next level?

IBM Watson IoT Platform Journey

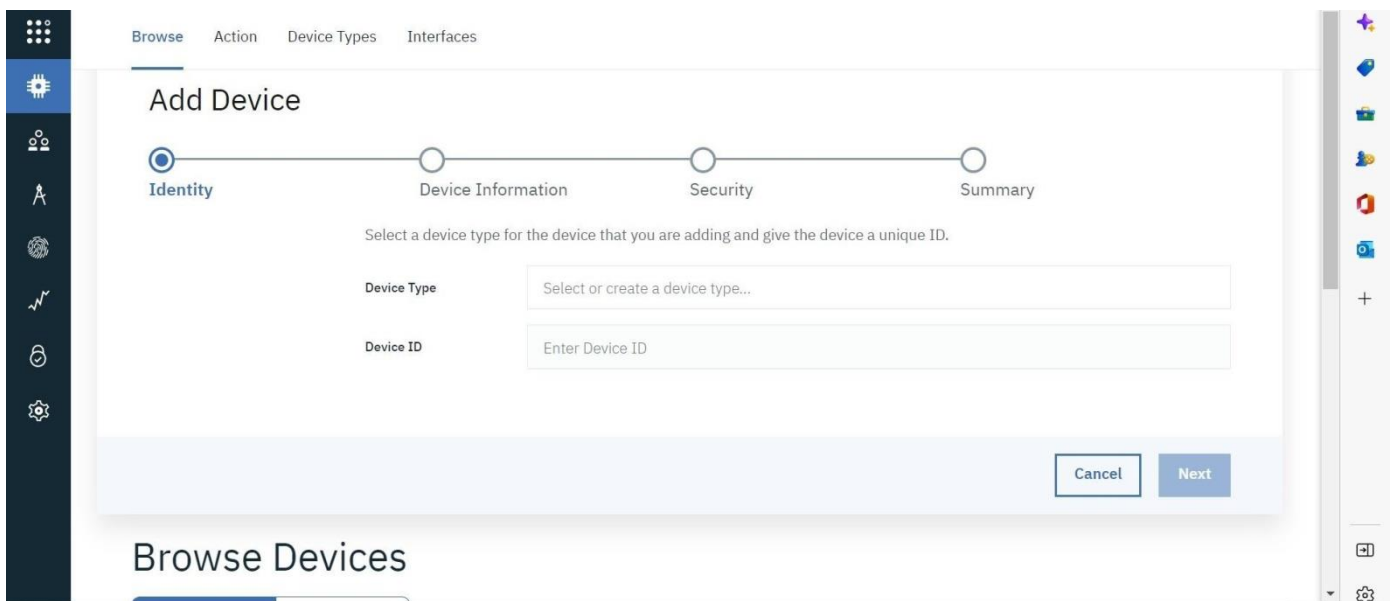
☒ Lite

☐ Non-Production

- 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.

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

Browse Action Device Types Interfaces

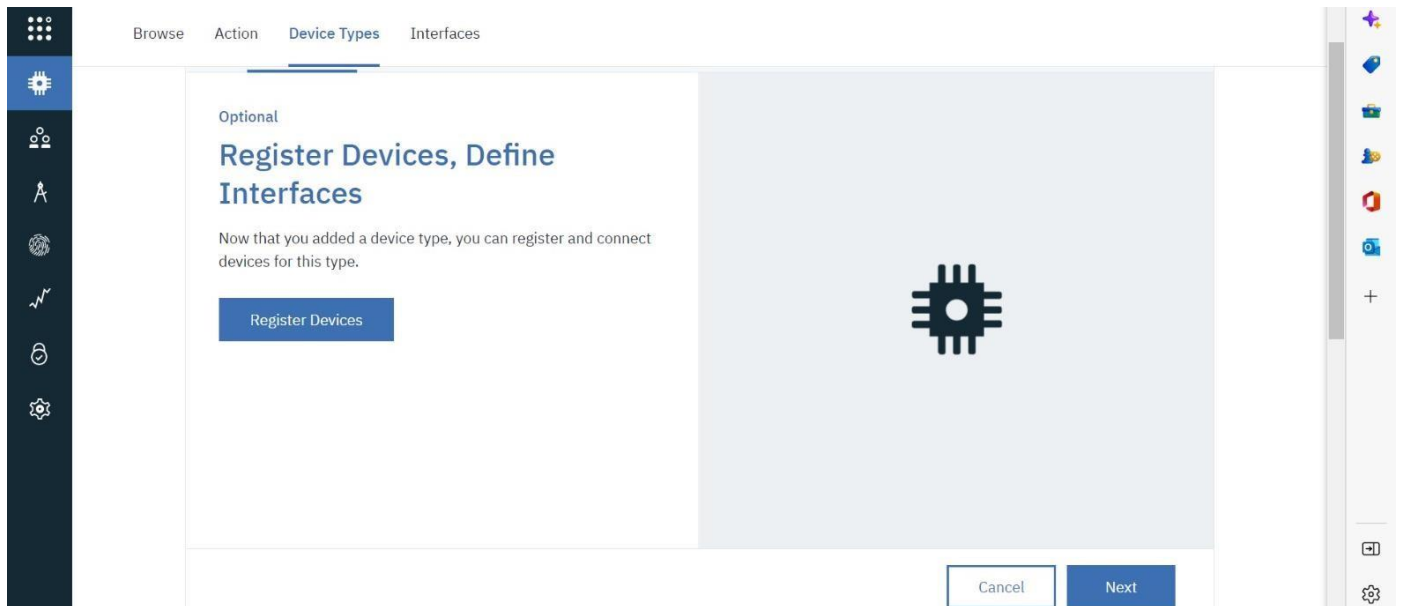
Add type

Identity Device Information

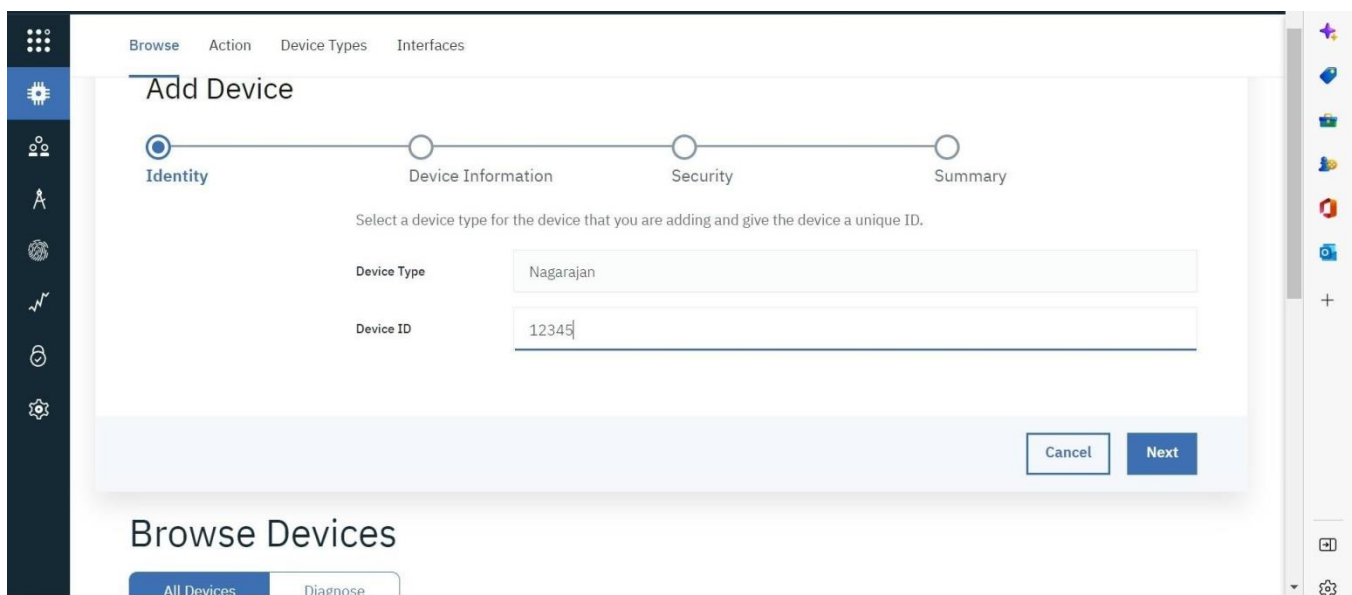
These attributes will be used as a template for new devices that are assigned this device type

Serial Number	<input type="text" value="Enter Serial Number"/>	Manufacturer	<input type="text" value="Enter Manufacturer"/>
Model	<input type="text" value="Enter Model"/>	Device Class	<input type="text" value="Enter Device Class"/>
Description	<input type="text" value="Enter Description"/>	Firmware Version	<input type="text" value="Enter Firmware Version"/>
Hardware Version	<input type="text" value="Enter Hardware Version"/>	Descriptive Location	<input type="text" value="Enter Descriptive Location"/>

- Click on Register Device.



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



➤ Click on Next

The screenshot shows the 'Add Device' form in the 'Device Information' step. The progress bar at the top indicates the current step. The form contains several input fields for device details and an 'Add Metadata' button.

Navigation: Browse | Action | Device Types | Interfaces

Progress: Identity (checked) | **Device Information** | Security | Summary

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

Serial Number	<input type="text" value="Enter Serial Number"/>	Manufacturer	<input type="text" value="Enter Manufacturer"/>
Model	<input type="text" value="Enter Model"/>	Device Class	<input type="text" value="Enter Device Class"/>
Description	<input type="text" value="Enter Description"/>	Firmware Version	<input type="text" value="Enter Firmware Version"/>
Hardware Version	<input type="text" value="Enter Hardware Version"/>	Descriptive Location	<input type="text" value="Enter Descriptive Location"/>

➤ Click on Next

The screenshot shows the 'Add Device' form in the 'Security' step. The progress bar at the top indicates the current step. The form provides instructions for selecting an authentication token and includes an input field for a self-provided token.

Navigation: Browse | Action | Device Types | Interfaces

Progress: Identity (checked) | Device Information (checked) | **Security** | Summary

Instructions: 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

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.

- Click on Finish
-
- 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 search bar labeled 'Search by Device ID' is present. The main content area shows a table of devices, with one device (ID 12345) selected. Below the table, the 'Recent Events' tab is active, displaying a list of events with columns for Event, Value, Format, and Last Received. The events are simulated data points. The bottom status bar indicates '3 Simulations running'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Disconnected	abcd	Device	15 Oct 2022 9:47 PM	

Event	Value	Format	Last Received
event_1	{"randomNumber":54,"temp":65,"hum":89}	json	a few seconds ago
event_1	{"randomNumber":44,"temp":30,"hum":94}	json	a few seconds ago
event_1	{"randomNumber":27,"temp":29,"hum":87}	json	a few seconds ago
event_1	{"randomNumber":42,"temp":18,"hum":88}	json	a few seconds ago
event_1	{"randomNumber":43,"temp":43,"hum":91}	json	a few seconds ago

3 Simulations 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

```
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 20117 0 0:00:03 0:00:03 --:--:-- 48190
pi@raspberrypi:~$ sudo dpkg -i iot_1.0-2_armhf.deb
(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 20117 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
    Limit: ~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[2567]: 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
    100% |#####| done
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 1.0MB/s
Collecting requests-toolbelt==0.7.0 (from ibmiotf)
  Downloading requests-toolbelt-0.8.0-py2.py3-none-any.whl (54kB)
    100% |#####| 61kB 1.6MB/s
Collecting xmldict==0.10.2 (from ibmiotf)
  Downloading xmldict-0.11.0-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 (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 (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/bbc33ad957e82f7b71ba80e316d65a83d9d735a0d12e0c0418
Running setup.py bdist_wheel for dicttoxml ... done
Stored in directory: /home/pi/.cache/pip/wheels/45/52/59/96910b33ec6a7b2ae66a13765401b50def5468024078e12c0e
Running setup.py bdist_wheel for paho-mqtt ... done
Stored in directory: /home/pi/.cache/pip/wheels/20/d8/0d/acdc8f289011b7be7de71deebef0642fb3be0313dfff0493
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:

```
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 07:10:37,765 ibmiotf.device.Client INFO Connected successfully: d:gegtl4:mydevice:mydevice
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
SensorData Invalid
Published Temperature = 28 C Humidity = 50 % to IBM Watson
Published Temperature = 28 C Humidity = 50 % to IBM Watson
Published Temperature = 28 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.

The screenshot shows the IBM Watson IoT Platform interface. The 'Browse Devices' page displays a table of devices. The table has columns for Device ID, Status, Device Type, Class ID, Date Added, and Descriptive Location. Two devices are listed: '12345' and 'abcd_1', both with a 'Disconnected' status and a 'Wifi' symbol. The page also includes a search bar, a 'Device Simulator' toggle, and a '3 Simulations running' notification.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Disconnected	abcd	Device	15 Oct 2022 9:47 PM	
abcd_1	Disconnected	abcd	Device	16 Oct 2022 1:29 PM	

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

The screenshot displays the IBM Watson IoT Platform dashboard. The top navigation bar includes tabs for 'Browse', 'Action', 'Device Types', and 'Interfaces'. A search bar labeled 'Search by Device ID' is present. The main content area shows a table of devices. The selected device (ID: 12345) is in a 'Disconnected' state. Below the device list, a 'Recent Events' tab is active, showing a stream of data events. The events table has columns for 'Event', 'Value', 'Format', and 'Last Received'. The events are JSON objects containing random numbers, temperature, and humidity data. A status bar at the bottom indicates '3 Simulations running'.

Event	Value	Format	Last Received
event_1	{"randomNumber":54,"temp":65,"hum":89}	json	a few seconds ago
event_1	{"randomNumber":44,"temp":30,"hum":94}	json	a few seconds ago
event_1	{"randomNumber":27,"temp":29,"hum":87}	json	a few seconds ago
event_1	{"randomNumber":42,"temp":18,"hum":88}	json	a few seconds ago
event_1	{"randomNumber":43,"temp":43,"hum":91}	json	a few seconds ago

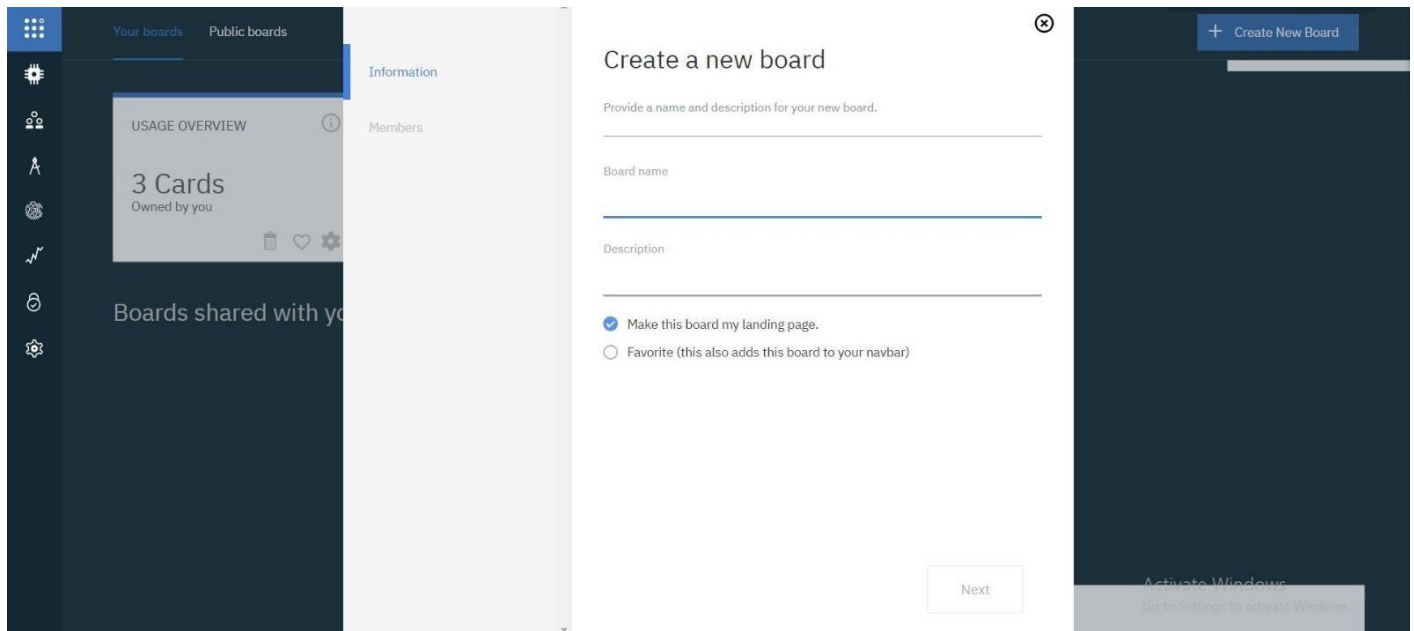
Step-4: Creating boards and cards for visualization of data:

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

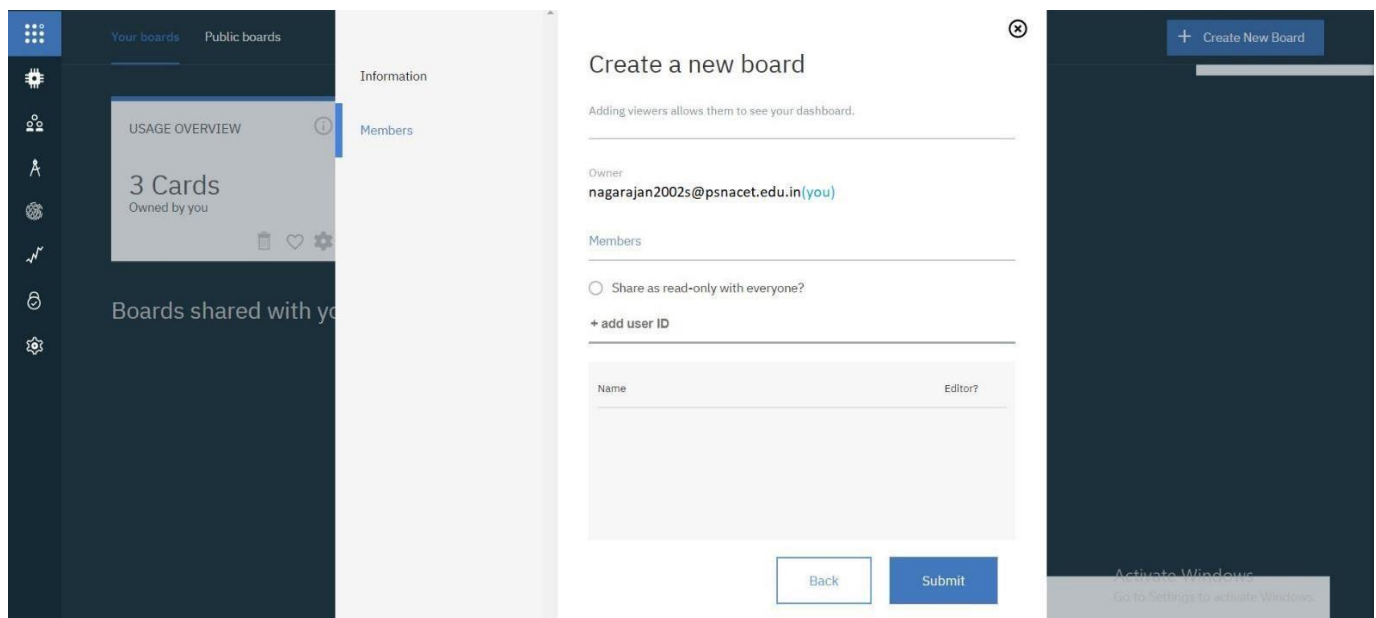
The screenshot shows the 'Your boards' page in the IBM Watson IoT Platform. The page has a dark theme and a sidebar with navigation icons. The main content area displays two boards: 'USAGE OVERVIEW' with 3 cards and 'RISK AND SECURITY OVERVIEW' with 4 cards. A large dashed box with a plus sign indicates where to create a new board. Below these boards, there is a section for 'Boards shared with you'. A status bar at the bottom indicates '1 Simulation running'. An 'Activate Windows' watermark is visible in the bottom right corner.

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

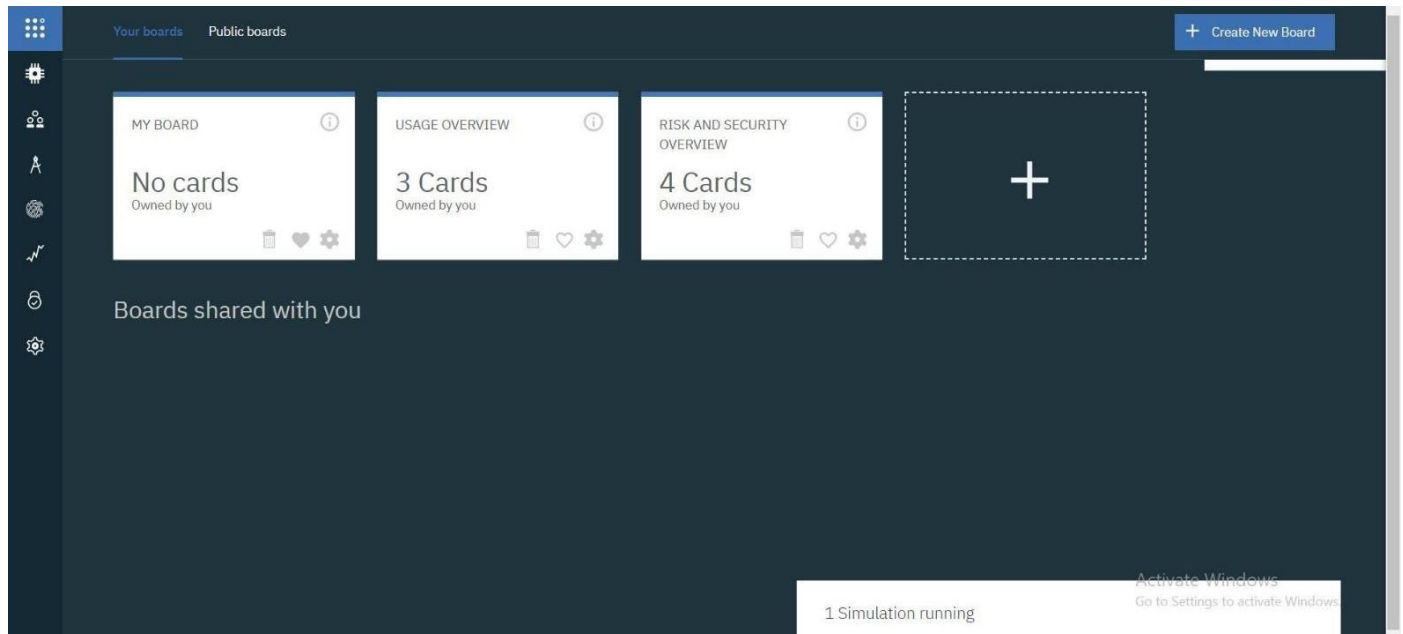
The given below window appears give a name and description to your board as shown in the window below.



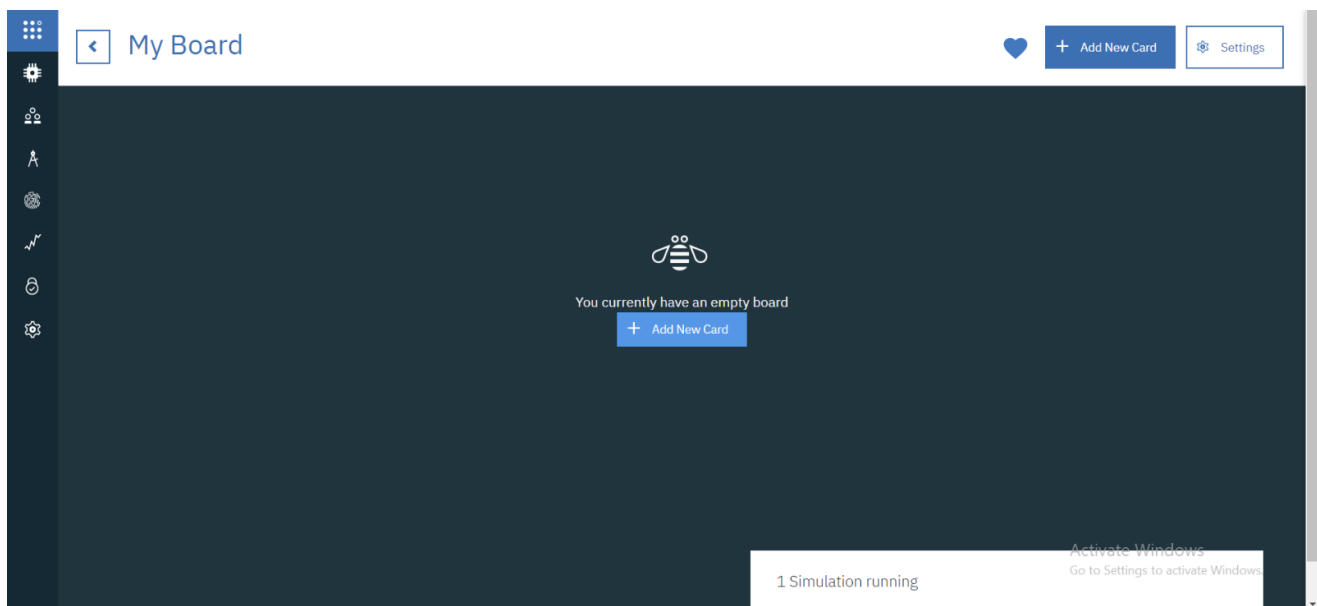
- Then click on Next you get the below window then again click on Submit



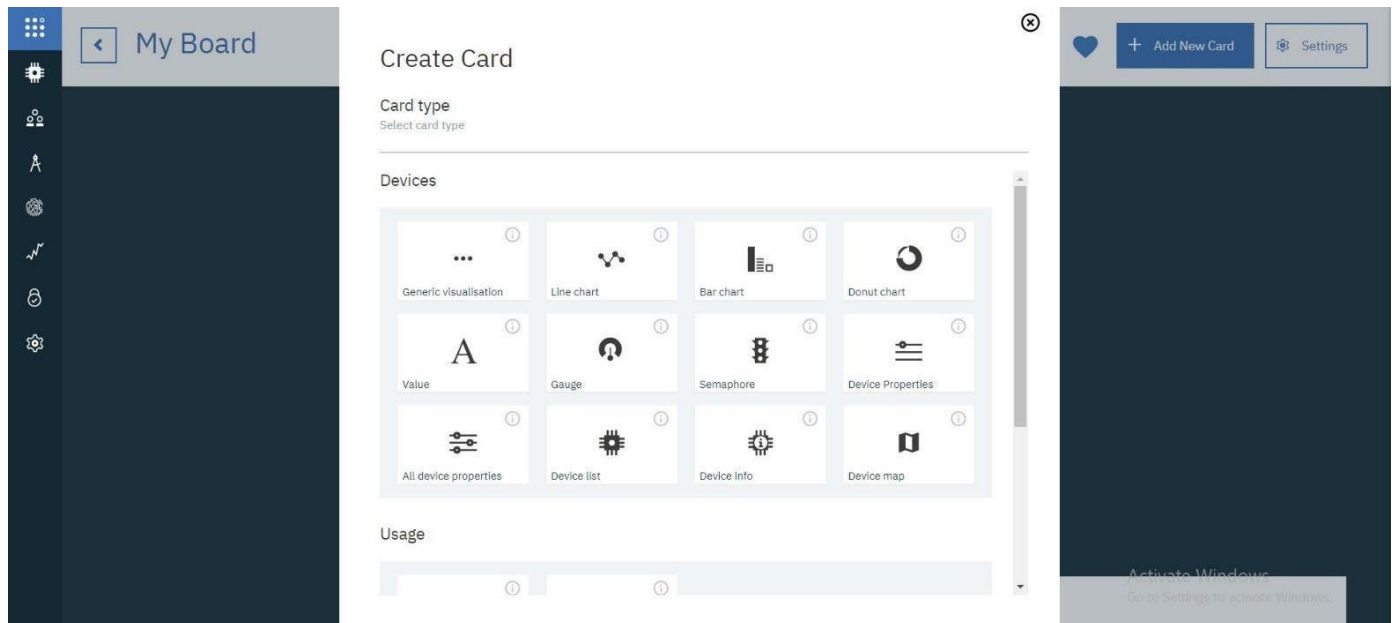
- Then double click on your boards name which you have created.



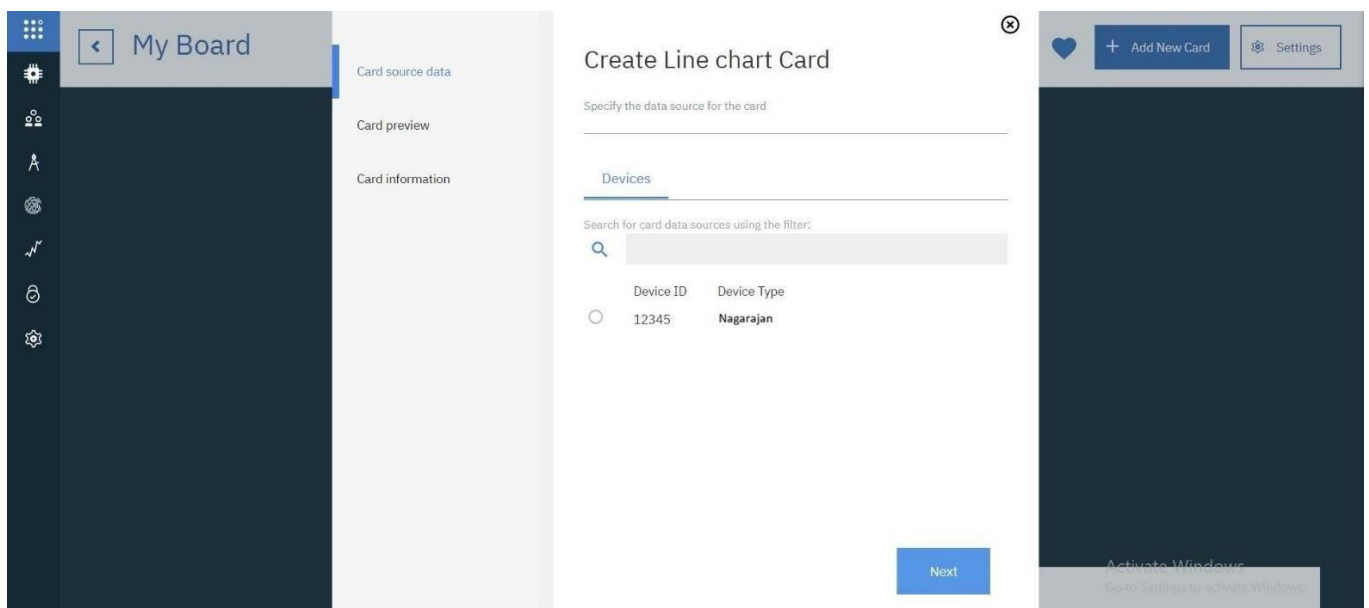
- Click on Add New Card



- 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

Temp & Hum

12345

Card source data

Card preview

Card information

Create Line chart Card

Connect data set

Temperature

Event

event_1

Property

Temperature

Name

Temperature

Type

Number

Unit

Max

100

Back

Next

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

Temp & Hum

12345

Card source data

Card preview

Card information

Create Line chart Card

Enter title and description of the card

Title

Line chart

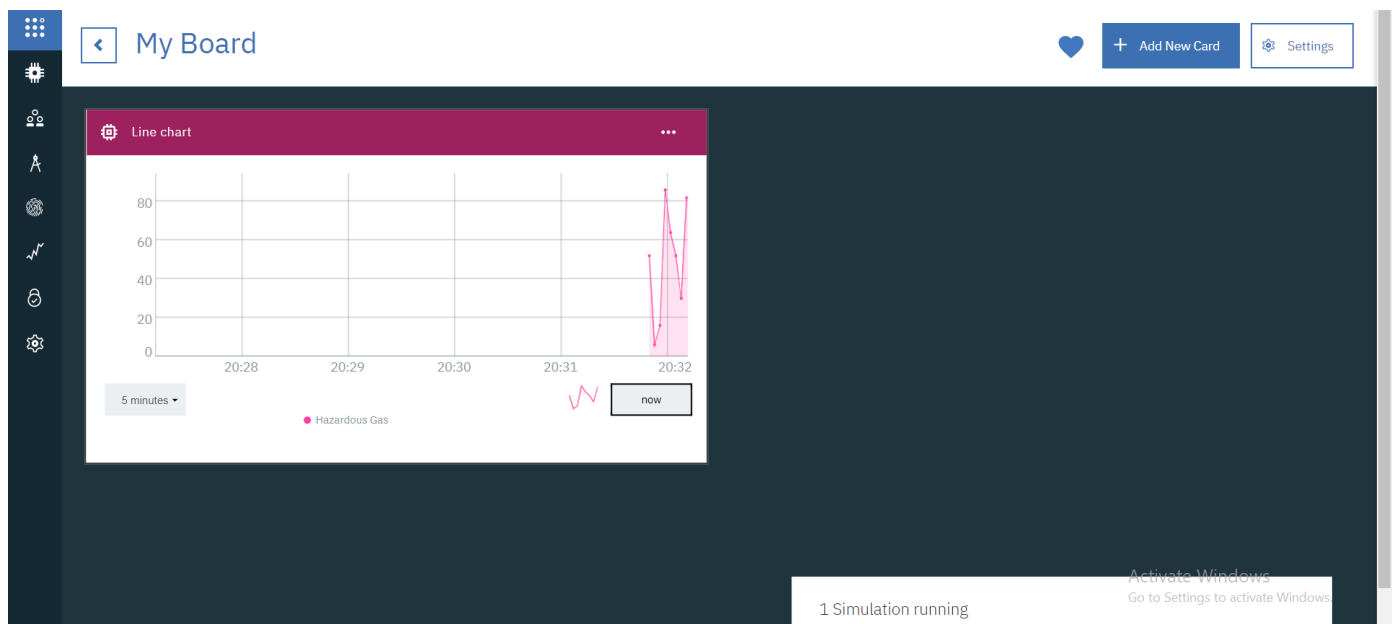
Color scheme

A line chart to display time series information with historic and live data.

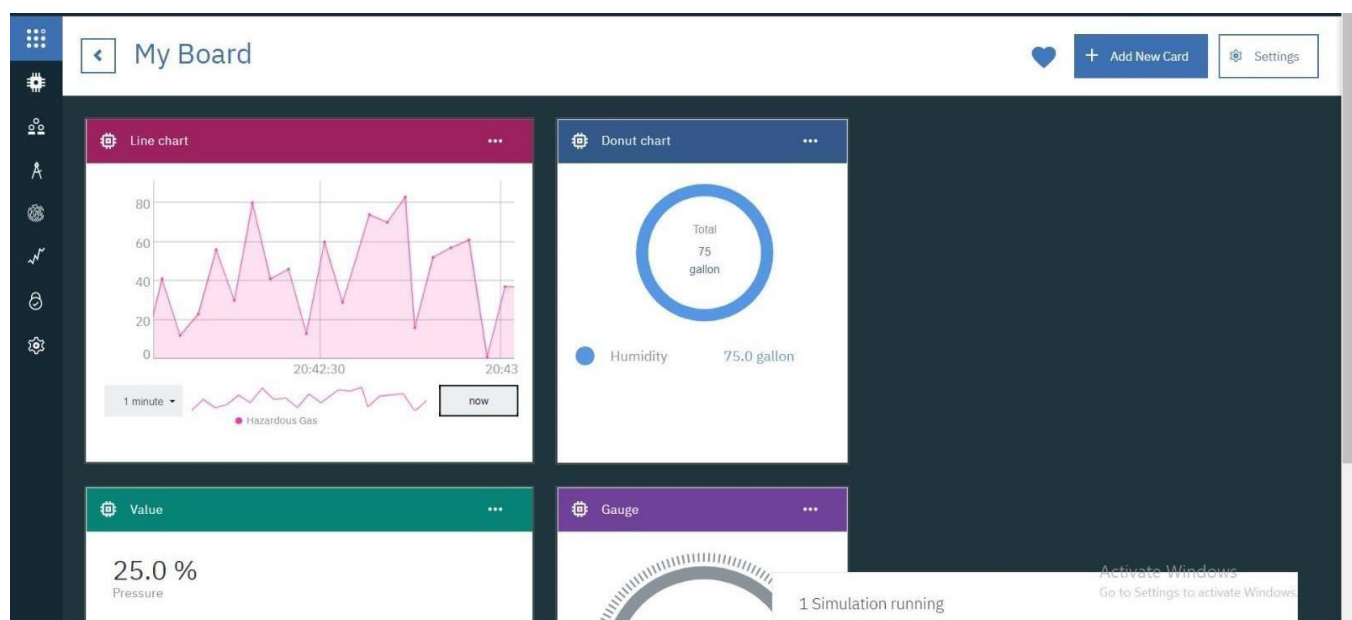
Back

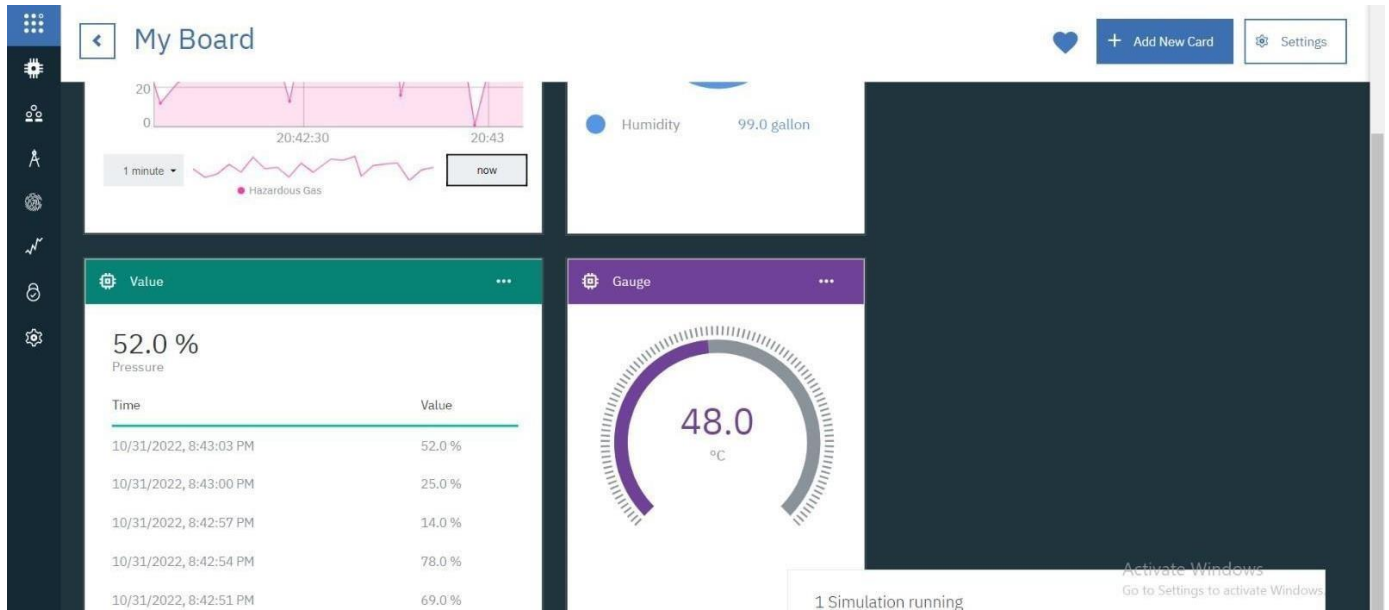
Submit

➤ 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.