

SENDING DATA FROM RASPBERRY-PI TO IBM WATSON

Date	10 NOVEMBER 2022
Team ID	PNT2022TMID01812
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

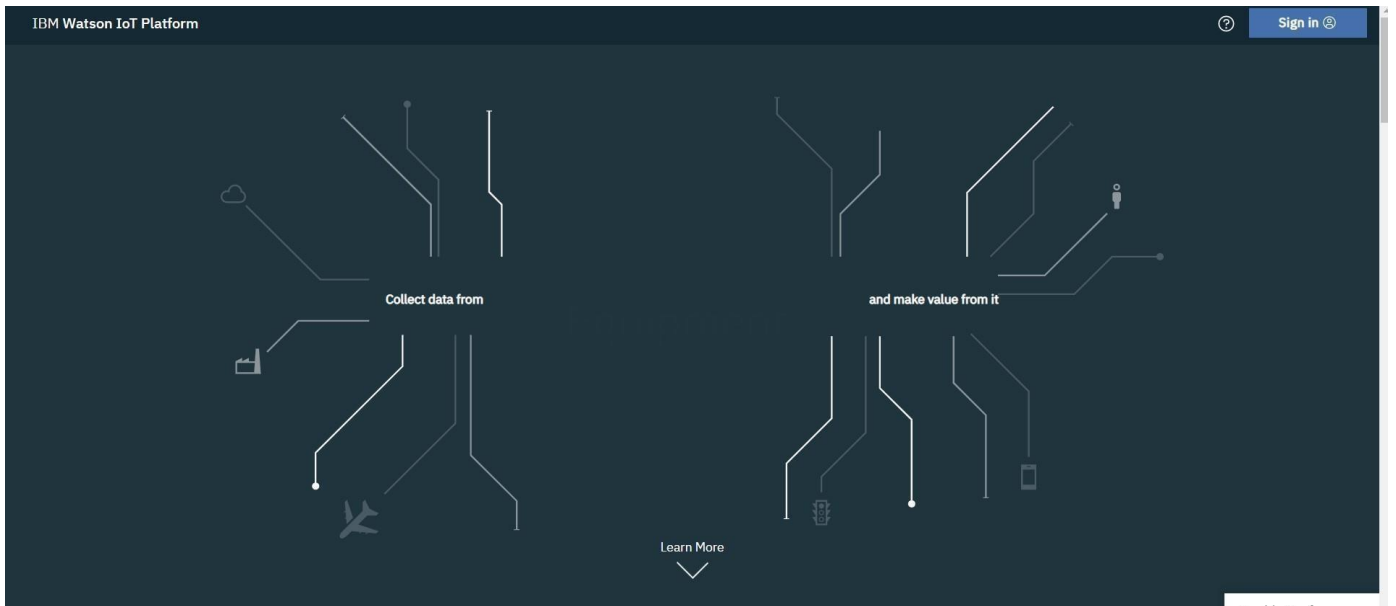
SOFTWARE:

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.



Log in to IBM

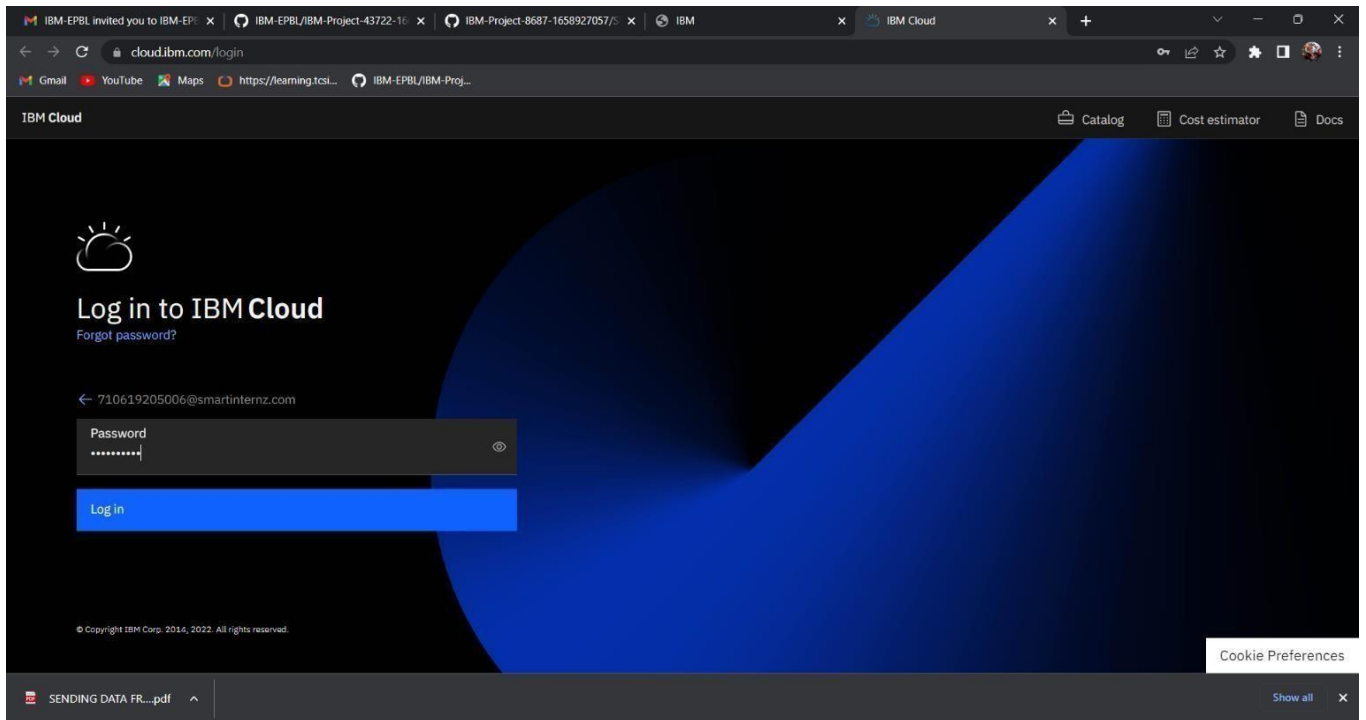
IBMId [Forgot IBMId?](#)

☒ Remember me [?](#)

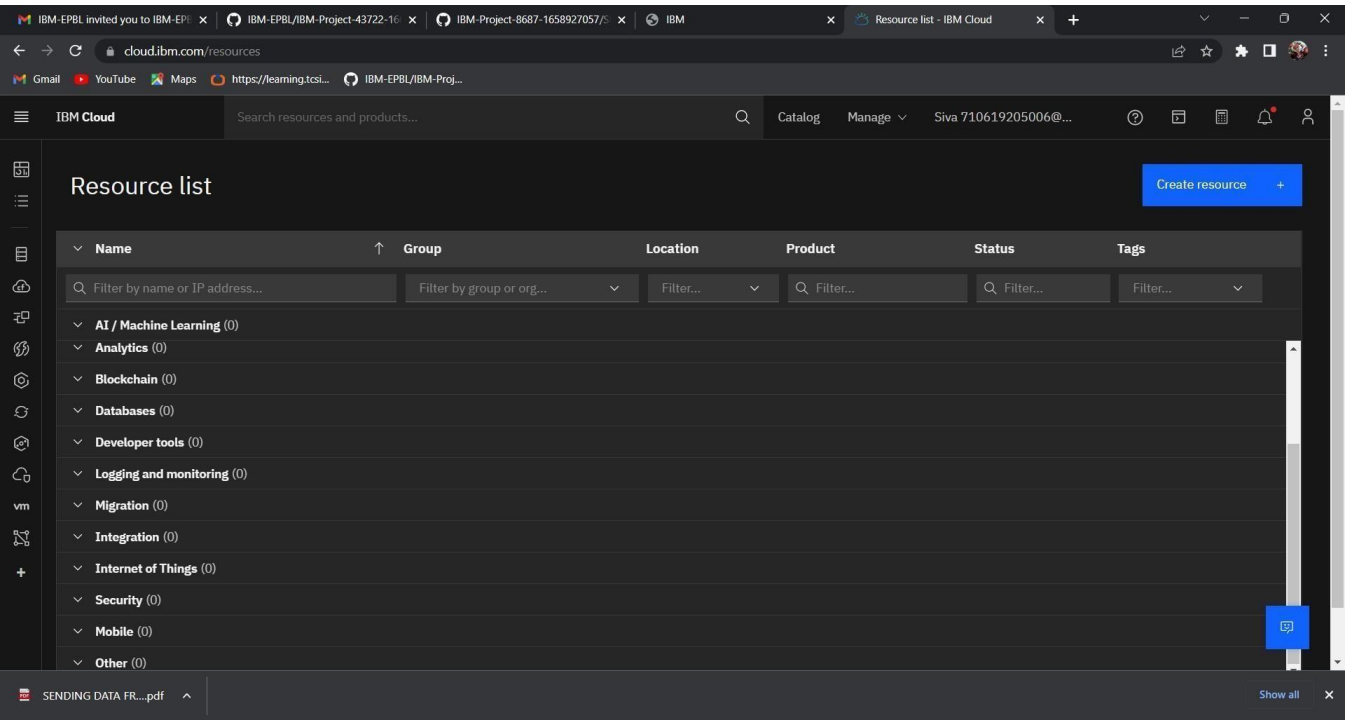
[Continue](#) →

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.

The screenshot shows the 'Create' page for the 'Internet of Things Platform' in the IBM Cloud catalog. The page is divided into a left sidebar, a main content area, and a right summary panel.

Left Sidebar:

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

Main Content Area:

- Create** (active) | About
- 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

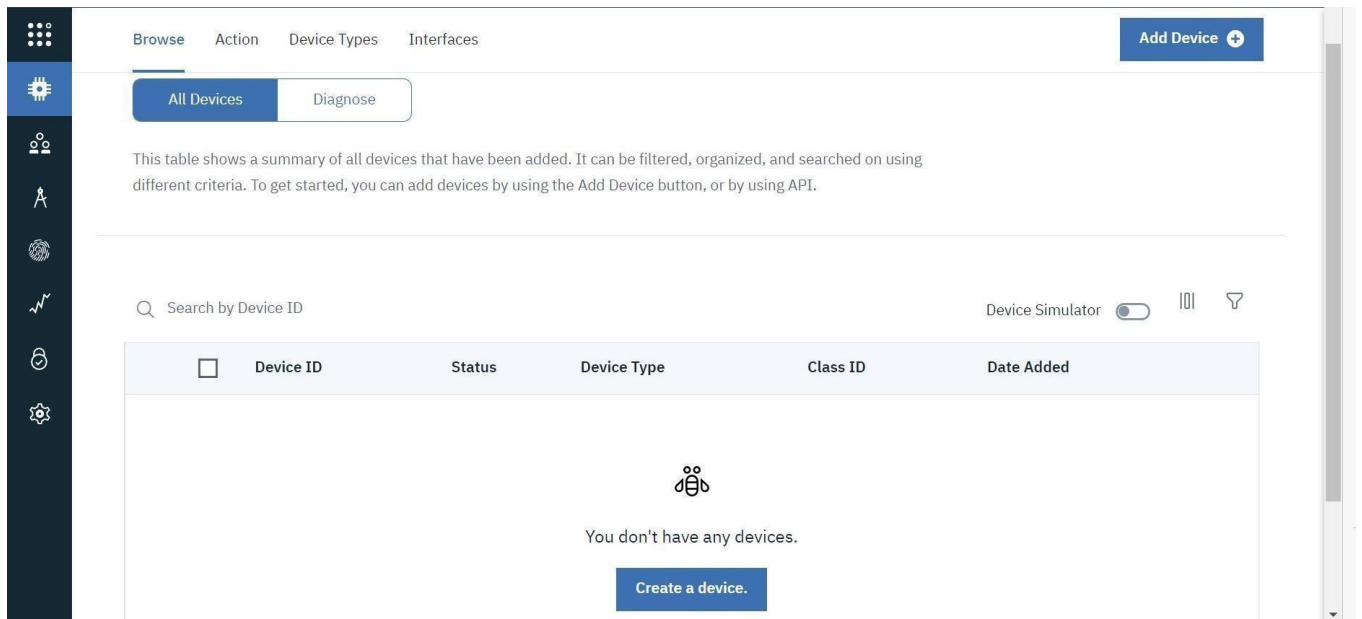
Right Summary Panel:

- 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** (button)
- Add to estimate (button)

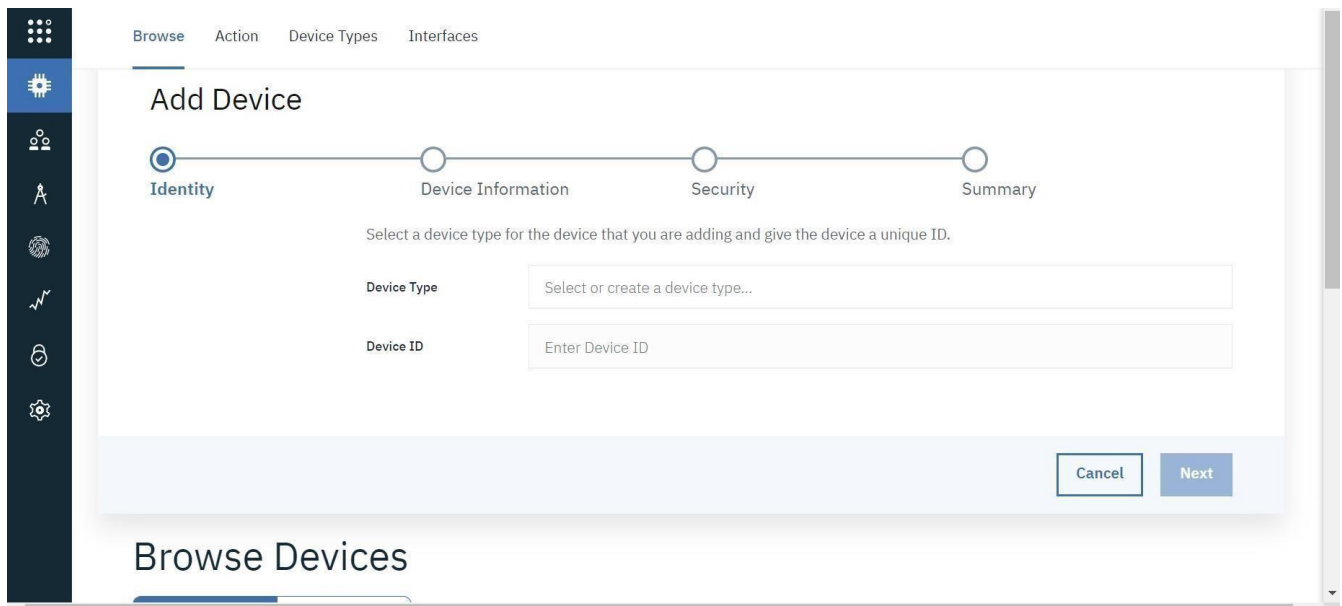
click on Launch

The screenshot shows the 'Internet of Things Platform-0g' dashboard. The top bar includes 'Resource list /', the resource name, status (Active), and actions (Add tags, Details, Actions...). The left sidebar has 'Manage' (selected), Plan, and Connections. The main content area features a large graphic of a central square with four 'U' shaped connectors, and text: '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.' Below this is a 'Launch' button and a 'Docs' button. Further down, a section titled 'Ready for the next level?' shows the 'IBM Watson IoT Platform Journey' with two progress indicators: 'Lite' (checked) and 'Non-Production' (unchecked). A blue chat icon is in the bottom right corner.

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

Device

Or

Gateway

Name

Nagarajan

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

Cancel

Next

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

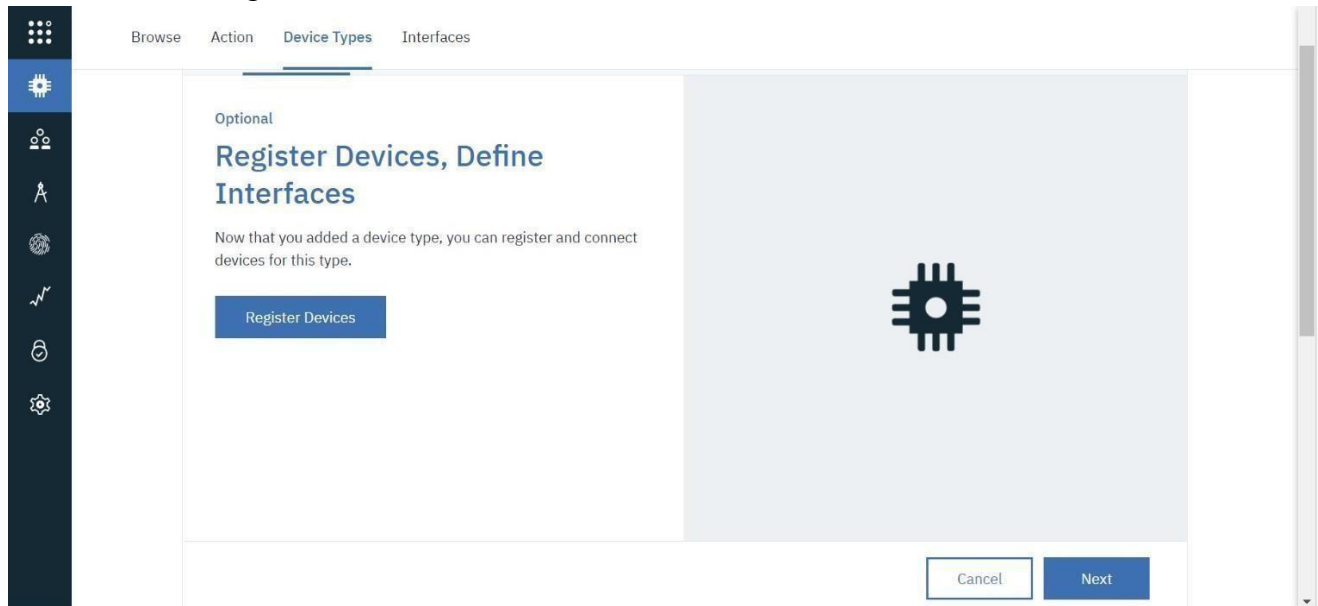
Descriptive Location

Enter Descriptive Location

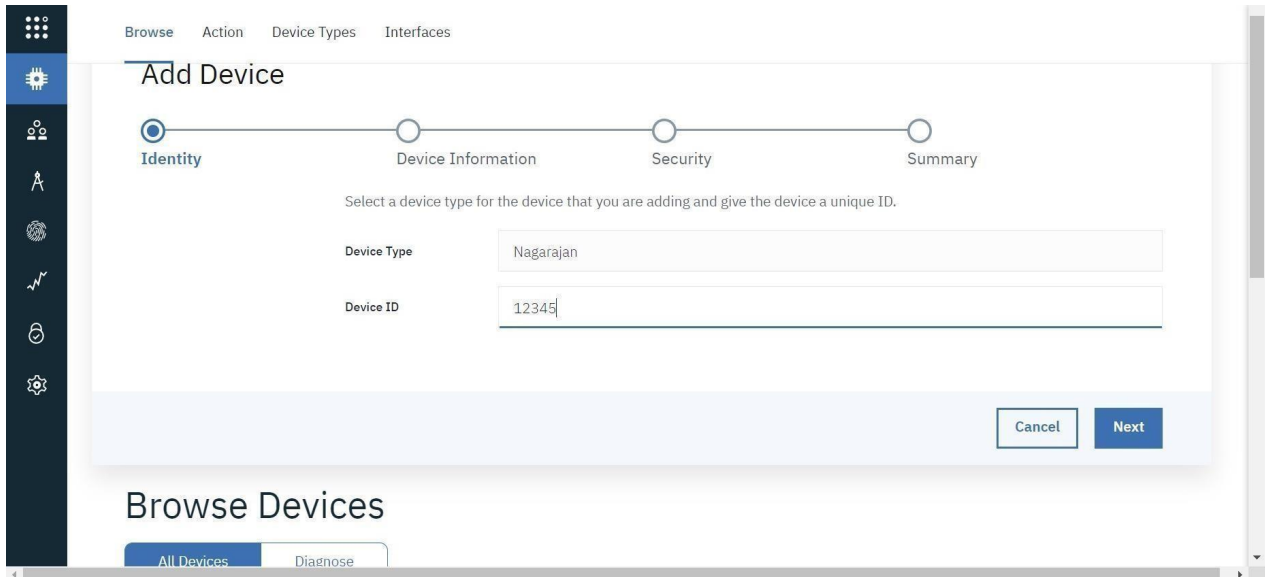
Back

Finish

Register Device.



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



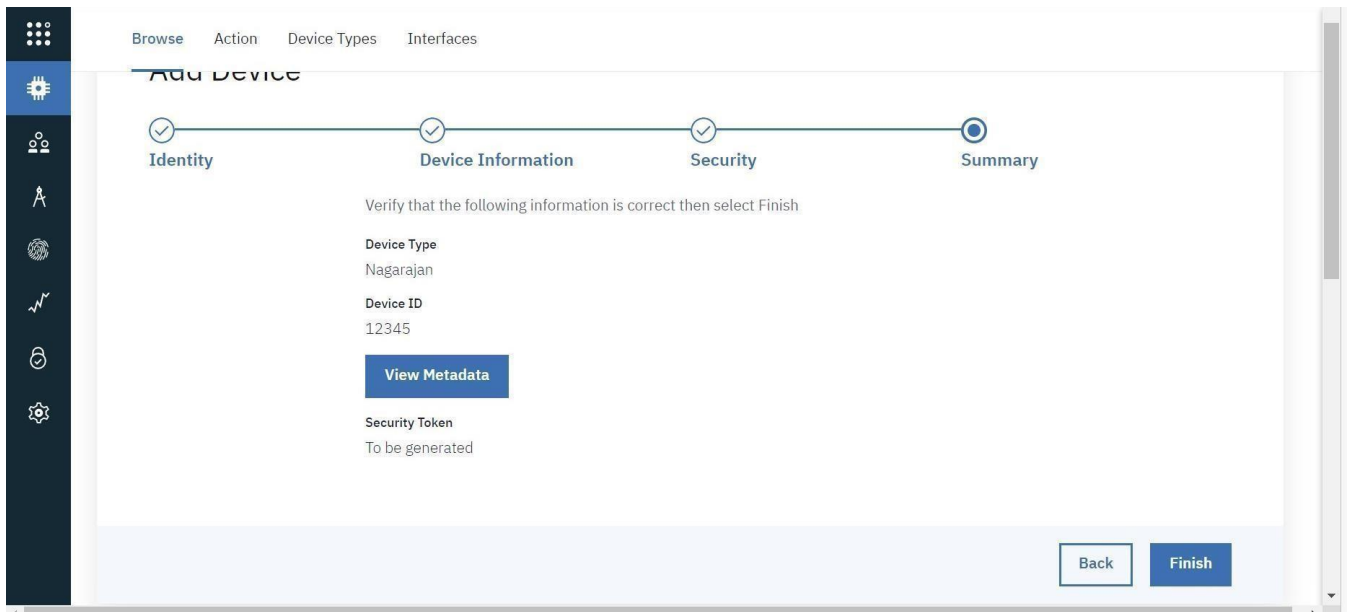
Click on Next

The screenshot shows the 'Add Device' form in a web application. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The left sidebar contains various icons. The main content area is titled 'Add Device' and features a progress bar with four steps: Identity (checked), Device Information (active), 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 includes 'Serial Number' (with a placeholder 'Enter Serial Number'), 'Model' (with a placeholder 'Enter Model'), 'Description' (with a placeholder 'Enter Description'), and 'Hardware Version' (with a placeholder 'Enter Hardware Version'). The right column includes 'Manufacturer' (with a placeholder 'Enter Manufacturer'), 'Device Class' (with a placeholder 'Enter Device Class'), 'Firmware Version' (with a placeholder 'Enter Firmware Version'), and 'Descriptive Location' (with a placeholder 'Enter Descriptive Location'). At the bottom left of the form is a button labeled 'Add Metadata' with a plus icon.

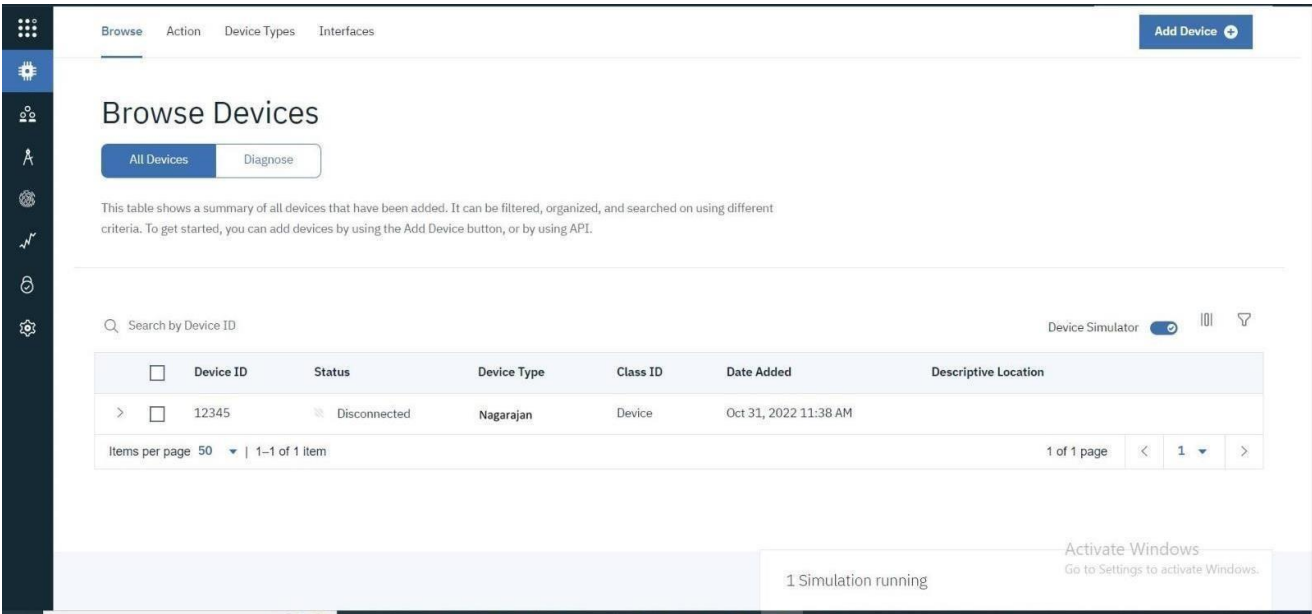
Click on Next

The screenshot shows the 'Add Device' form in a web application, specifically the 'Security' step. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The left sidebar contains various icons. The main content area is titled 'Add Device' and features a progress bar with four steps: Identity (checked), Device Information (checked), Security (active), and Summary. Below the progress bar, a message states: 'There are two options for selecting a device authentication token.' The form is divided into two columns. The left column is titled 'Auto-generated authentication token (default)' and contains the text: '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.' The right column is titled 'Self-provided authentication token' and contains the text: '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.' Below these columns is a text input field labeled 'Authentication Token' with a placeholder 'Enter an optional token' and an information icon. Below the input field, there is a note: 'Make a note of the generated token. Lost authentication tokens cannot be recovered. Tokens are encrypted before being stored.' and a statement: 'Authentication token are encrypted before we store them.'

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

```
File Edit Tabs Help
--2017-10-23 06:55:22-- http://ftp.nl.debian.org/debian/pool/main/o/openssl/lib
ssl1.0.0_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]

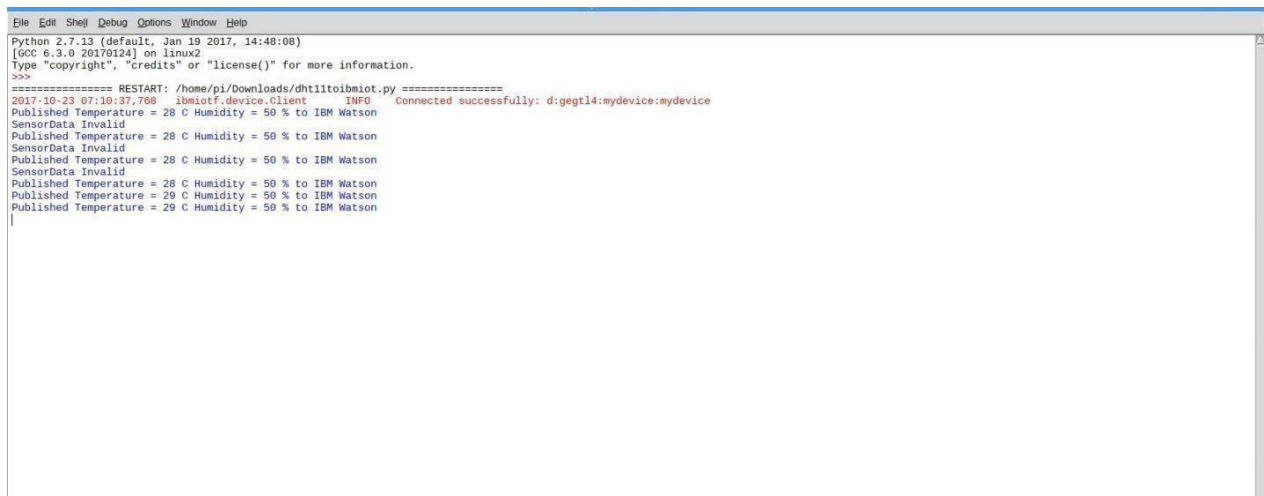
pi@raspberrypi:~$ sudo dpkg -i libssl1.0.0_1.0.1t-1+deb8u6_armhf.deb
Selecting previously unselected package libssl1.0.0:armhf.
(Reading database ... 115006 files and directories currently installed.)
Preparing to unpack libssl1.0.0_1.0.1t-1+deb8u6_armhf.deb ...
Unpacking libssl1.0.0:armhf (1.0.1t-1+deb8u6) ...
Setting up libssl1.0.0:armhf (1.0.1t-1+deb8u6) ...
pi@raspberrypi:~$ curl -LO https://github.com/ibm-messaging/iot-raspberrypi/rele
ases/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
100 104 0 104 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 29117 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)
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[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
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 xmltodict<0.10.2 (from ibmiotf)
  Downloading xmltodict-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-2.8-py2.py3-none-any.whl (58kB)
    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 (133kB)
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/bbc33ad957e82f7b71ba80e216d05a309d735a0d12e0c0418
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/d8/04/acdc8f289911b7be7de71deebef6642fb83be03134fff0493
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-2.8 iso8601-0.1.12 paho-mqtt-1.3.1 pytz-2017.2 requests-2.18.4 requests-toolbelt-0.8.0 urllib3-1.22 xmltodict-0.11.0
pi@raspberrypi:~$
```

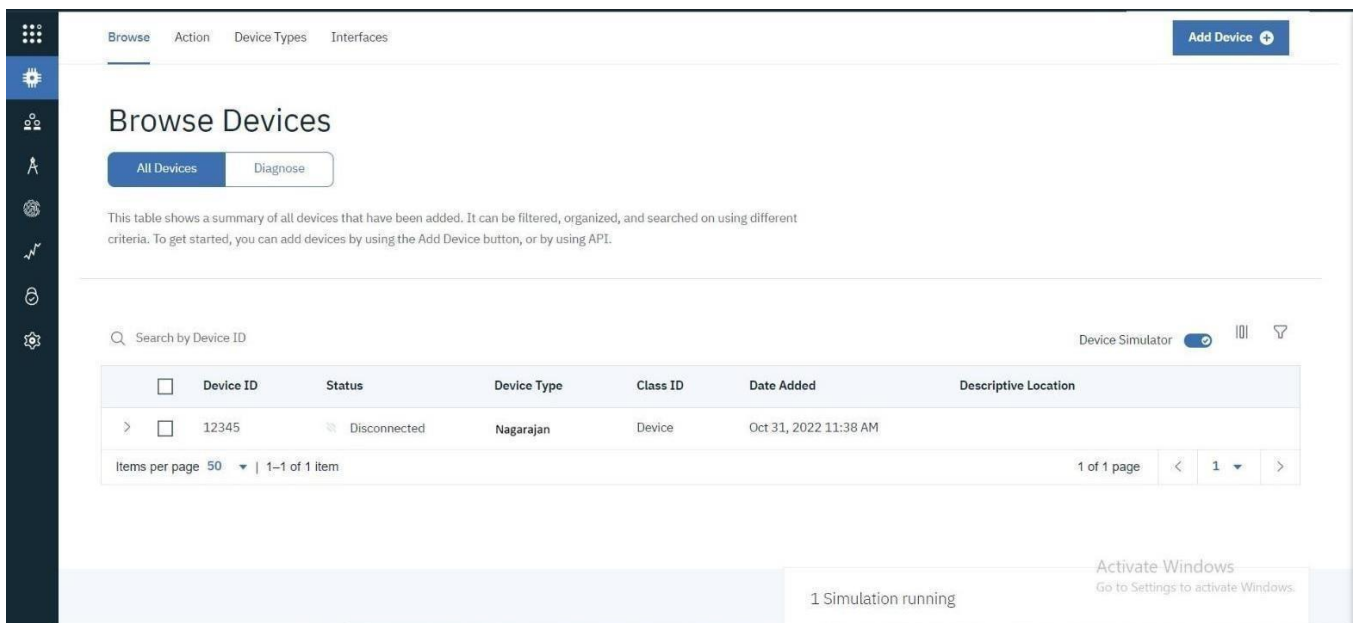
I have sent DHT-11 Sensors data to ibm bluemix .To get the code u need to login into IOT GYAN.
Then I get the image as follows in my pi's shell:



```
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/dht11toibmiot.py =====
2017-10-23 07:10:37,766 ibmiotf.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.



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

The screenshot displays the Watson IoT platform interface. On the left is a dark sidebar with navigation icons. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces', along with an 'Add Device' button. The main content area shows a table of devices. The selected device, ID 12345, is 'Disconnected' and of type 'Nagarajan'. Below the device header, there are tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is active, showing a list of events with columns for Event, Value, Format, and Last Received. The events are JSON payloads containing sensor data like 'Hazardous Gas', 'Temperature', and 'Humidity'. At the bottom right, there is a status bar indicating '1 Simulation running' and a Windows activation notice.

Event	Value	Format	Last Received
event_1	{"Hazardous Gas":61,"Temperature":88,"Humidit...	json	a few seconds ago
event_1	{"Hazardous Gas":20,"Temperature":36,"Humidit...	json	a few seconds ago
event_1	{"Hazardous Gas":79,"Temperature":56,"Humidit...	json	a few seconds ago
event_1	{"Hazardous Gas":52,"Temperature":82,"Humidit...	json	a few seconds ago
event_1	{"Hazardous Gas":26,"Temperature":33,"Humidit...	json	a few seconds ago

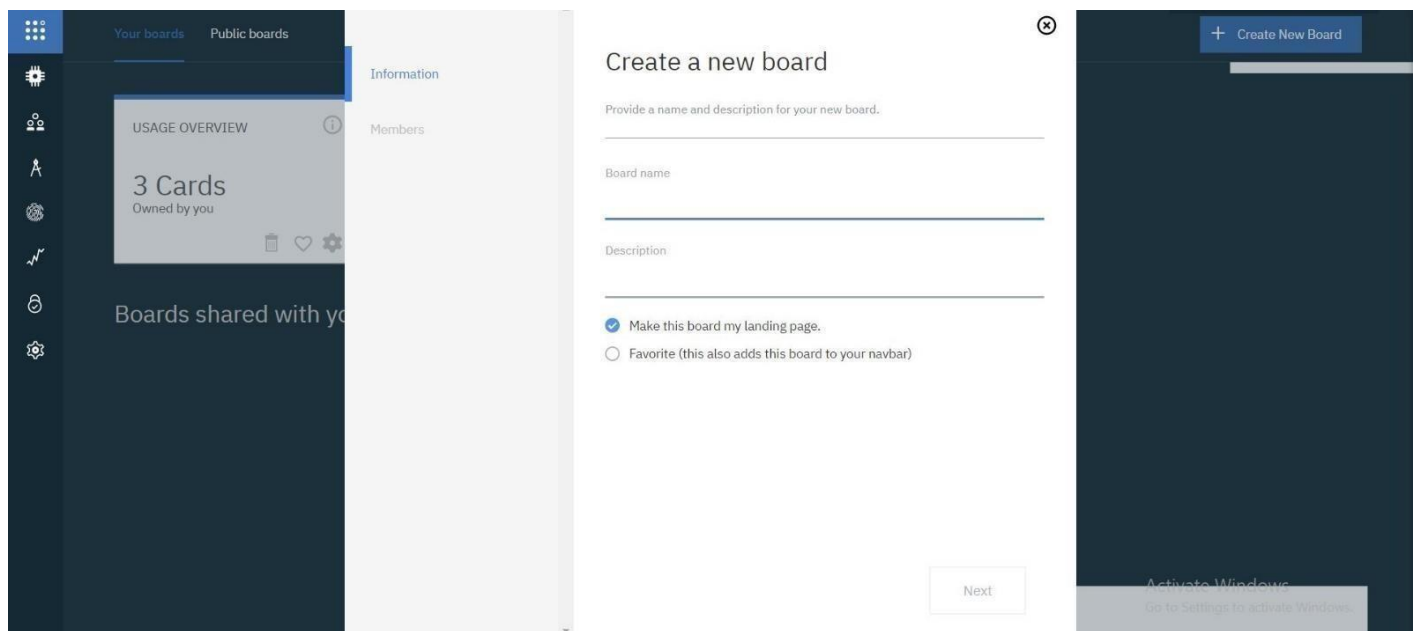
Step-4: Creating boards and cards for visualization of data:

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

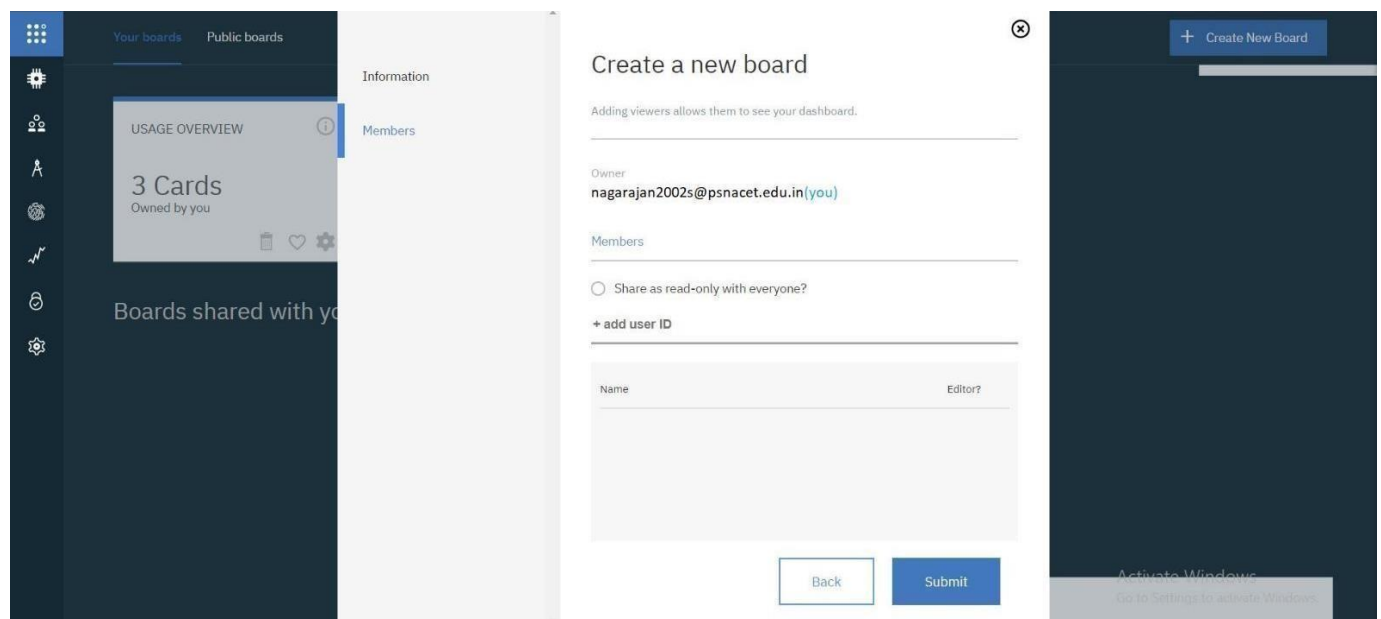
The screenshot shows the 'Your boards' section of the Watson IoT platform. The top navigation bar has 'Your boards' and 'Public boards' tabs, and a 'Create New Board' button. The main area displays two existing boards: 'USAGE OVERVIEW' with 3 cards and 'RISK AND SECURITY OVERVIEW' with 4 cards. Both boards are 'Owned by you' and have icons for deleting, favoriting, and configuring. To the right of these boards is a large dashed box with a plus sign, indicating where to click to create a new board. Below the boards, there is a section titled 'Boards shared with you'. At the bottom right, the same status bar from the previous screenshot is visible, showing '1 Simulation running' and a Windows activation notice.

Click on Create a new board to create a board .

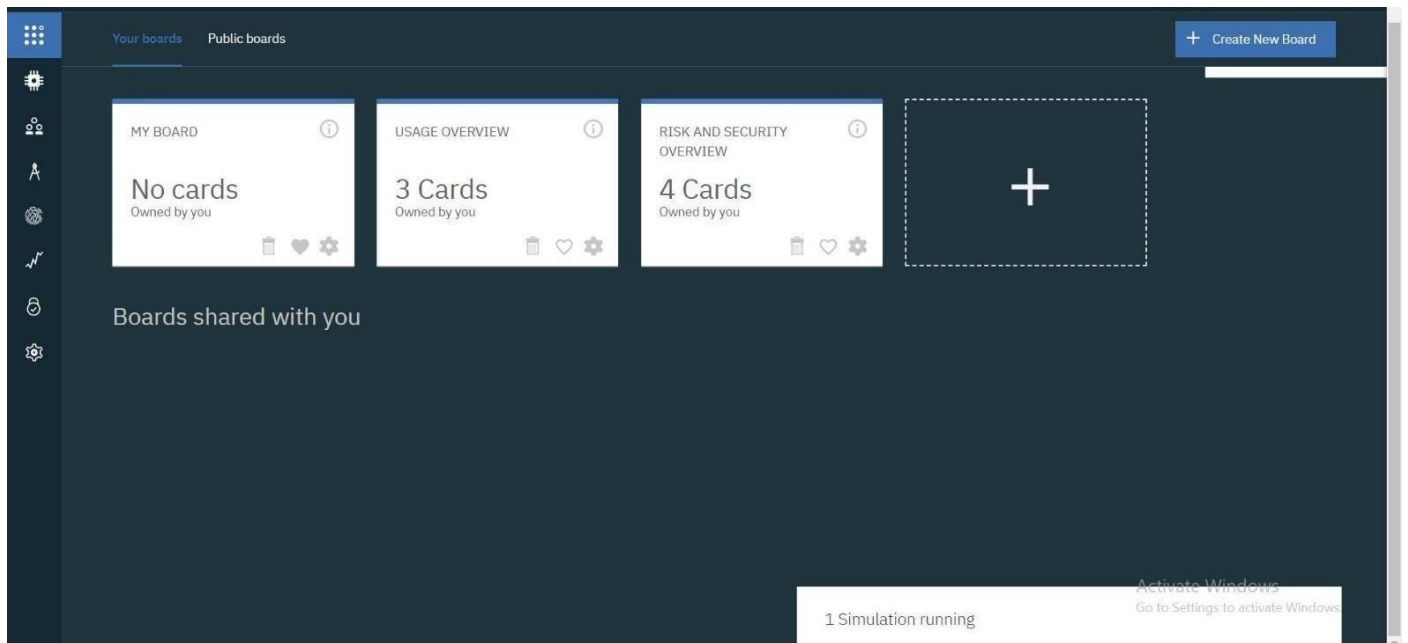
The given below window appears give a name and description to your board as shown in the window below.



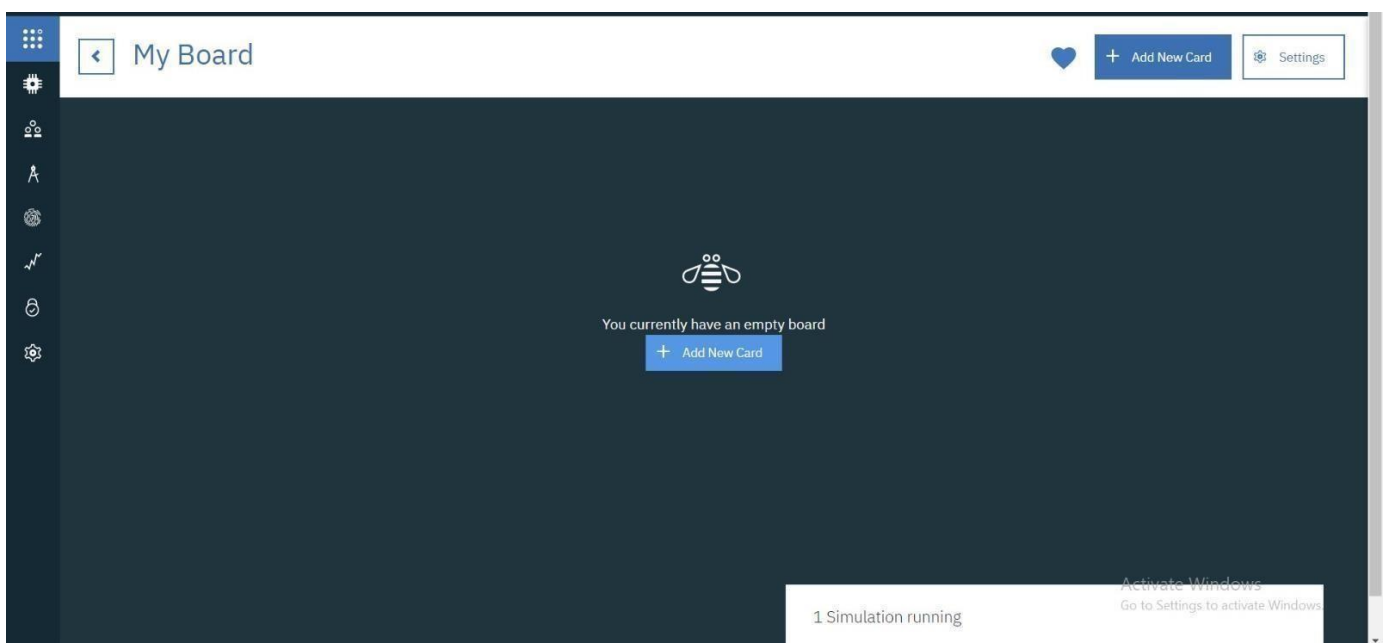
Then click on Next you get the below window then again click on Submit



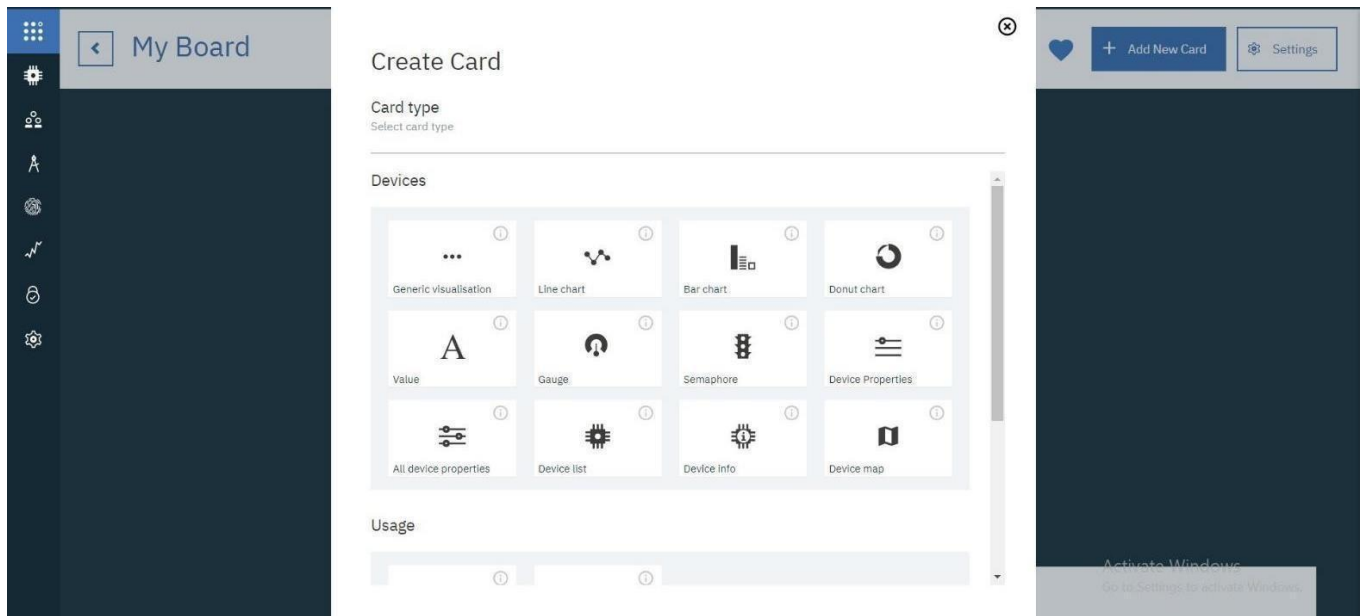
Then double click on your boards name which you have created.



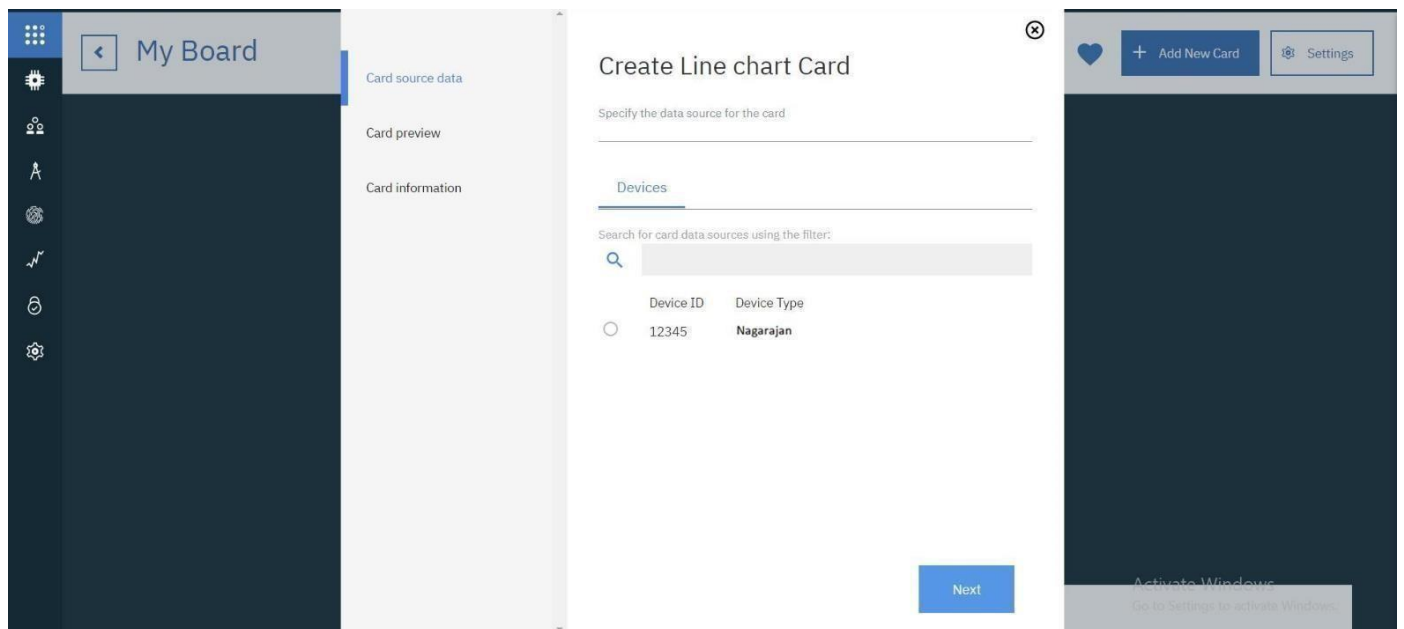
Click on Add New Card



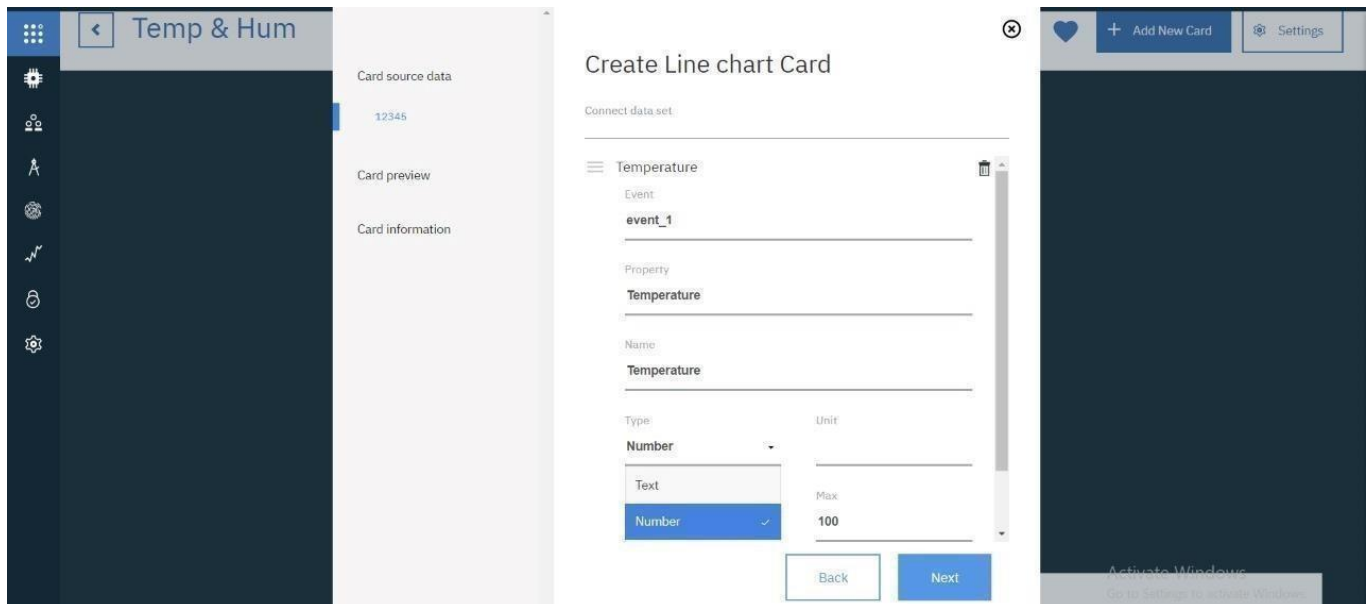
Select the type of Graph u want accordingly and click next



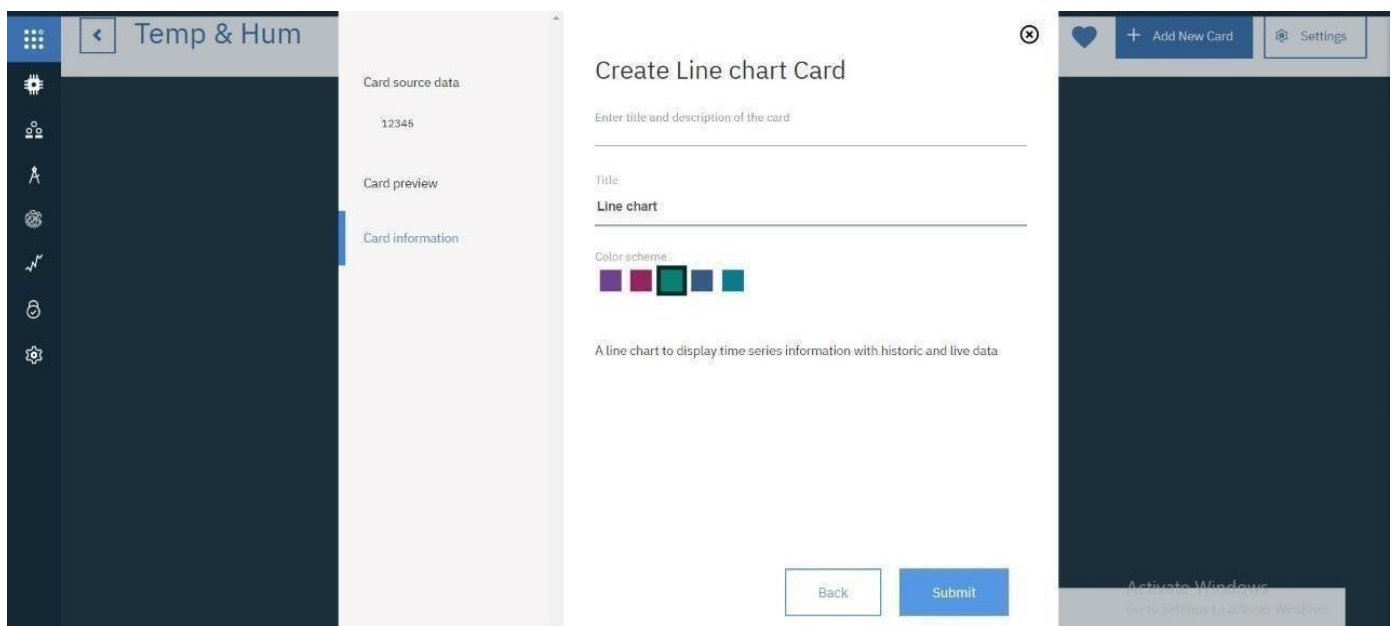
You get the below window, choose the Device and click on Next.



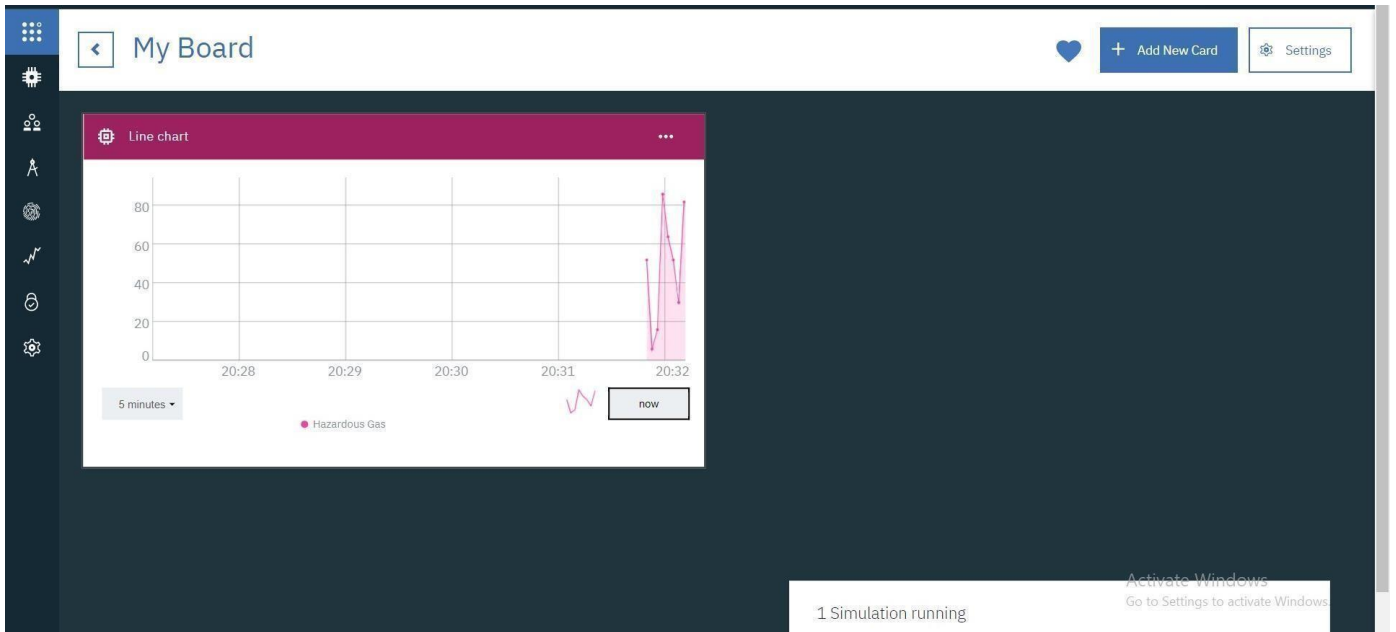
Select the event, properly to be visualized on your graph and click next. In my case it is humidity



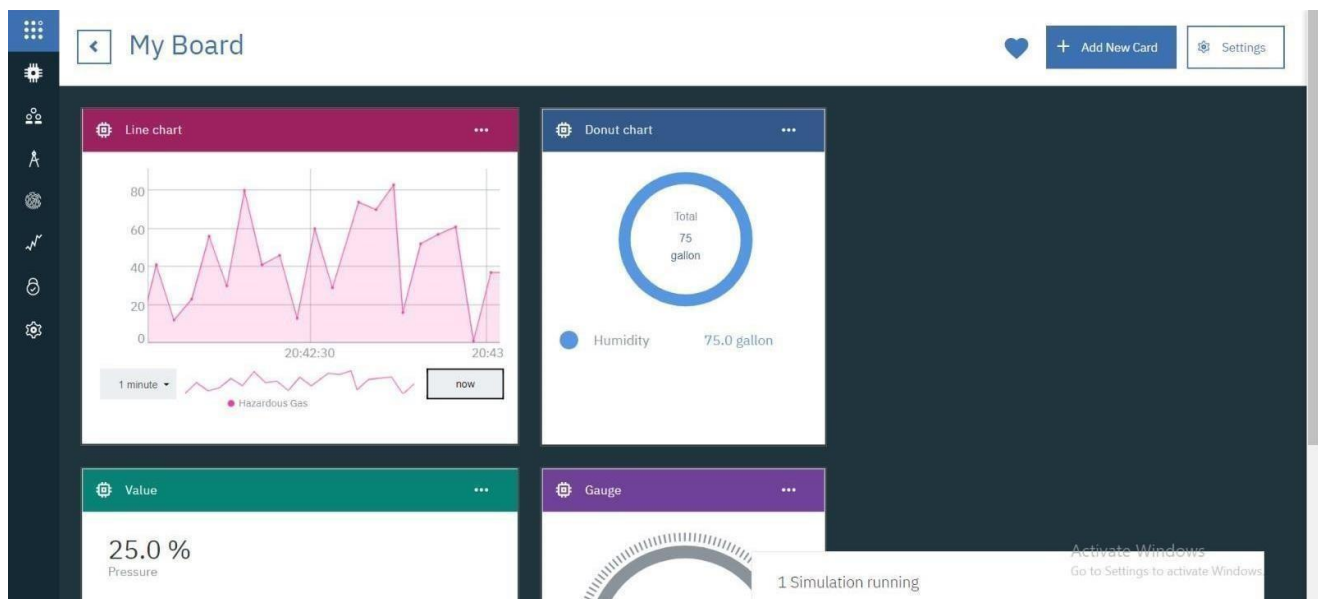
Then select the size of the graph and color of the graph board you want and click next

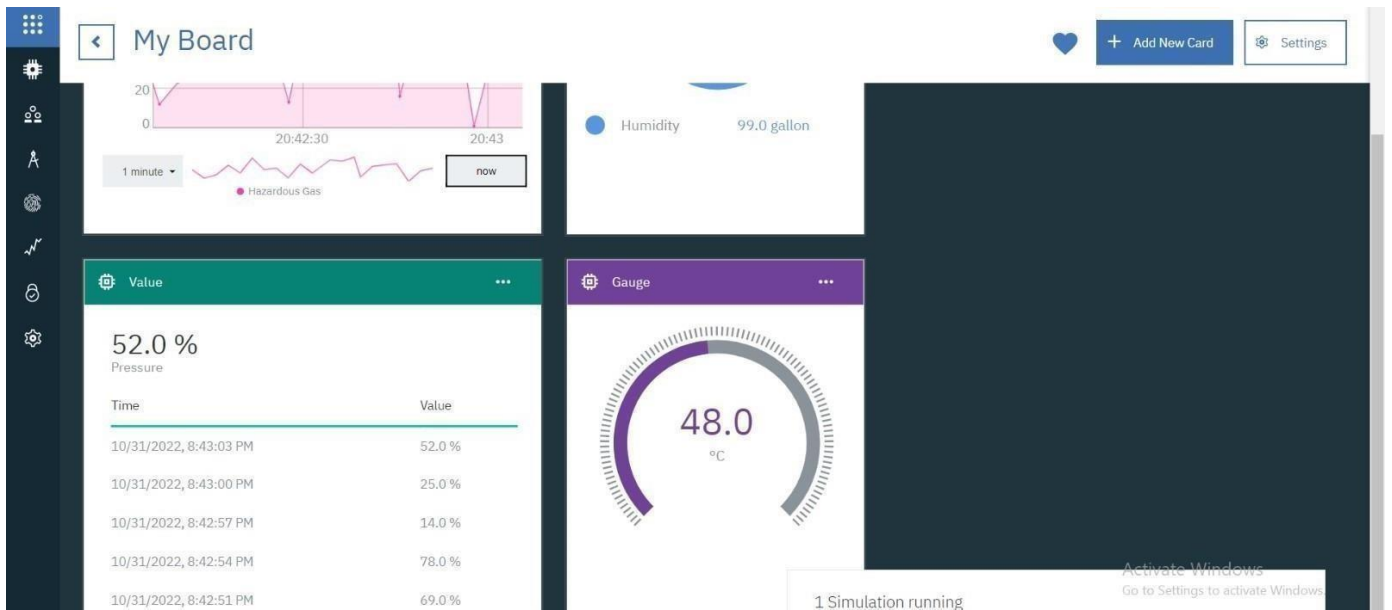


Here is the graph



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.