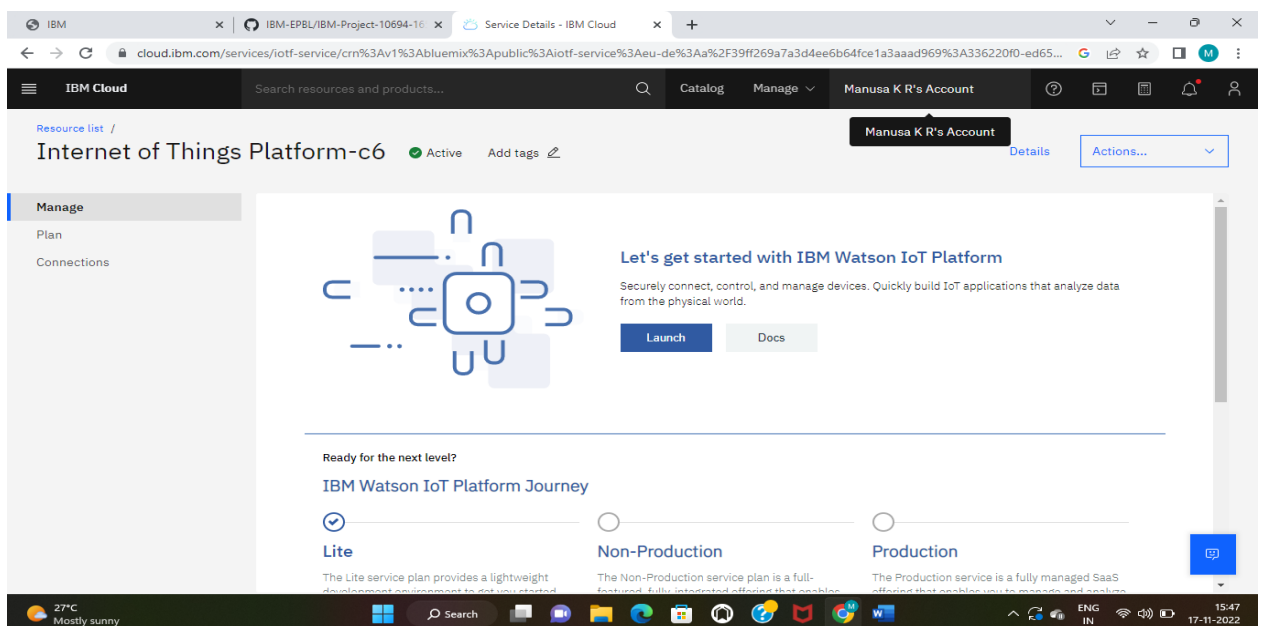
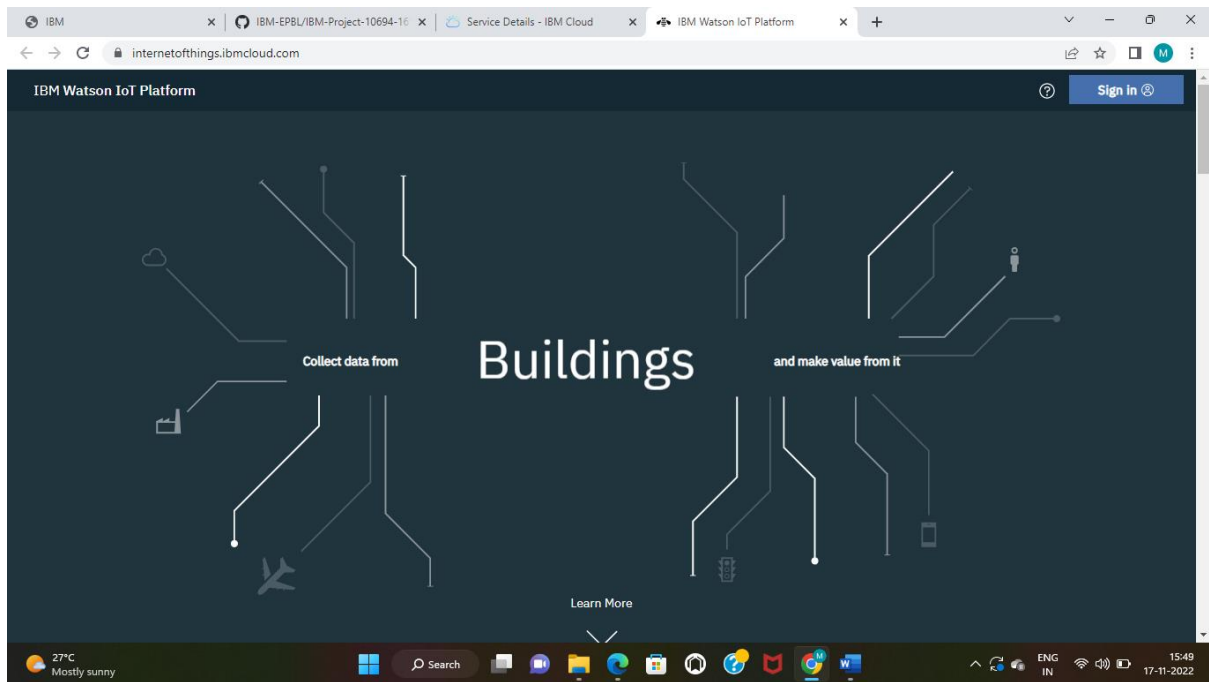


SENDING DATA FROM RASPBERRY-PI TO IBM WATSON

DATE	5 NOVEMBER
TEAM ID	PNT2022TMID15063
PROJECT TITLE	GAS LEAKAGE MONITORING AND ALERTING SYSTEM FOR INDUSTRIES

Step-1: Create a device in IBM Watson

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



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

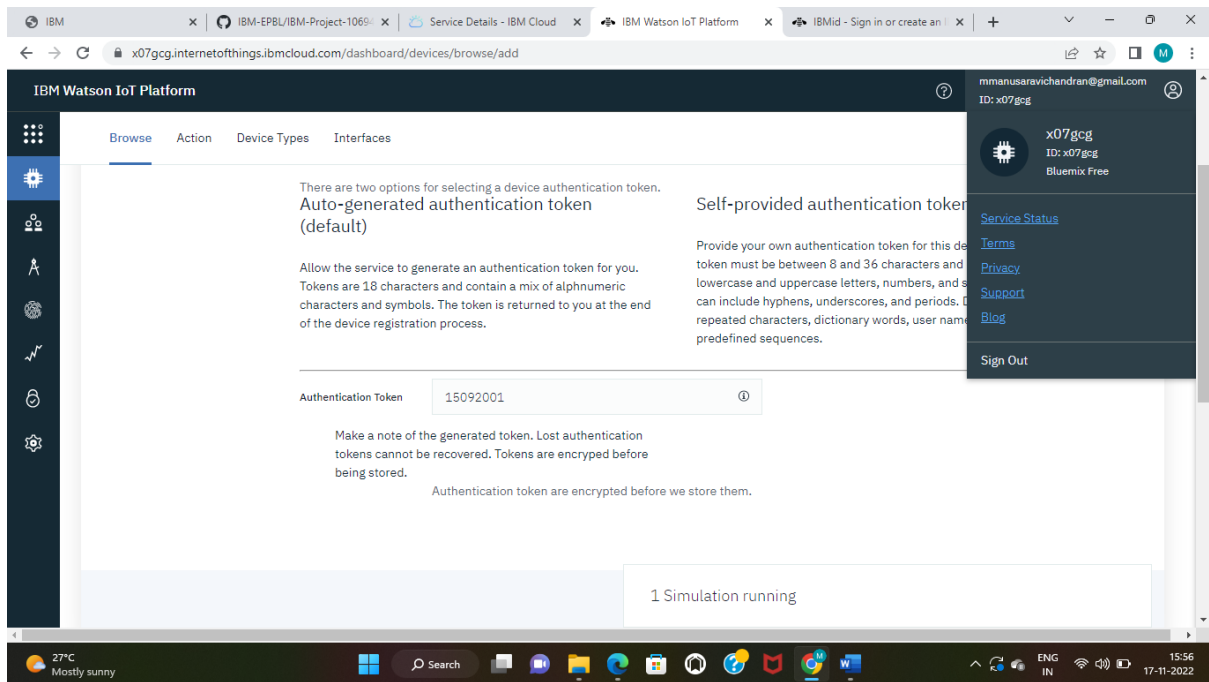
The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The main heading is 'Browse Devices', with tabs for 'All Devices' and 'Diagnose'. A table lists devices with columns: Device ID, Status, Device Type, Class ID, and Date Added. One device is listed with ID 123, status 'Disconnected', and type 'manusa'. A sidebar on the right shows the user profile for 'mmanusaravichandran@gmail.com' and links for 'Service Status', 'Terms', 'Privacy', 'Support', and 'Blog'. The bottom of the page shows a taskbar with various application icons and system status indicators.

Device ID	Status	Device Type	Class ID	Date Added
123	Disconnected	manusa	Device	Nov 16, 2022 10:57 PM

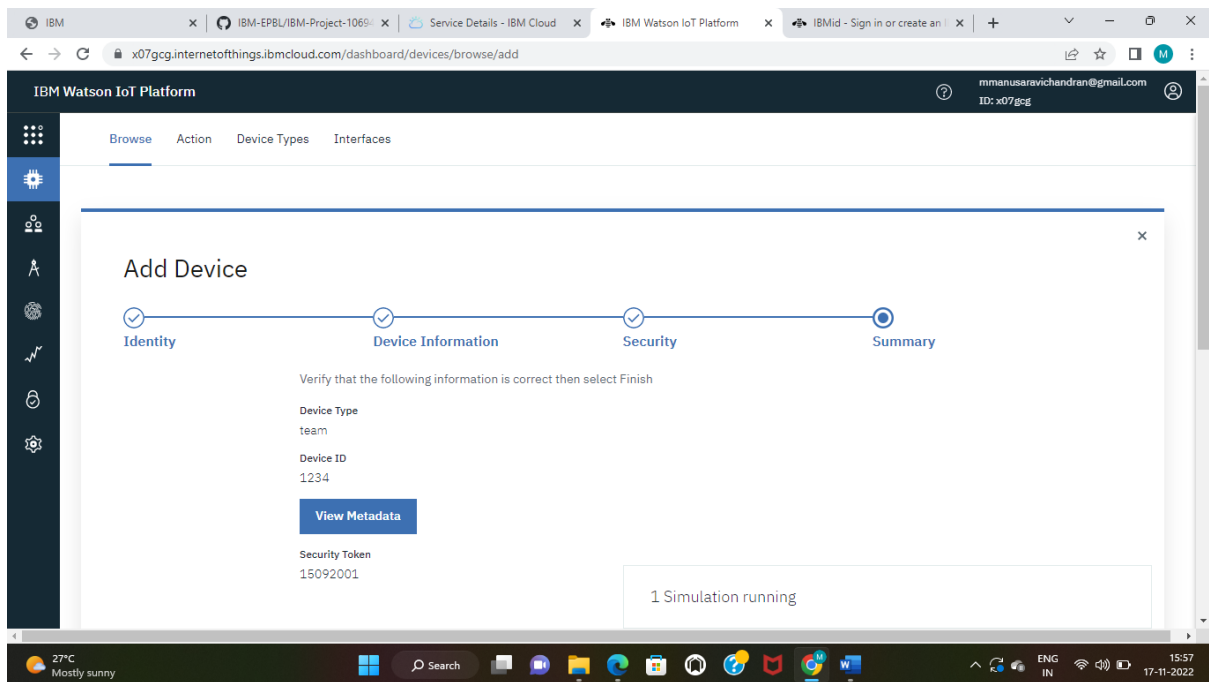
- After click on Add device this page will open

The screenshot shows the 'Add Device' page in the IBM Watson IoT Platform. The page has a progress bar with four steps: 'Identity', 'Device Information', 'Security', and 'Summary'. The 'Identity' step is currently active. Below the progress bar, there are input fields for 'Device Type' (containing 'team') and 'Device ID' (containing '1234'). At the bottom right, there are 'Cancel' and 'Next' buttons. The page also features a 'Browse Devices' section at the bottom with 'All Devices' and 'Diagnose' tabs. The bottom of the page shows a taskbar with various application icons and system status indicators.

- Go to device type and fill the details.



➤ Click on Finish



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`

```
File Edit Tabs Help
~2017-10-23 06:55:22~ http://ftp.nl.debian.org/debian/pool/main/o/openssl/lib
1.0.2.1-1.0-1-i-debdeb_armhf.deb
Resolving ftp.nl.debian.org (ftp.nl.debian.org)... 130.89.149.21, 2901:67c:2b64:
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: 'libssl1.0.0-1.0-1-i-debdeb_armhf.deb'

libssl1.0.0-1.0-1- 100%[=====] 847.61K 358KB/s in 2.4s

2017-10-23 06:55:25 (358 KB/s) - 'libssl1.0.0-1.0-1-i-debdeb_armhf.deb' saved [
867950/867950]

pi@raspberrypi:~$ sudo dpkg -i libssl1.0.0-1.0-1-i-debdeb_armhf.deb
Selecting previously unselected package libssl1.0.0-armhf.
(Reading database ... 115626 files and directories currently installed.)
Preparing to unpack libssl1.0.0-1.0-1-i-debdeb_armhf.deb ...
Unpacking libssl1.0.0-armhf (1.0.0-1-i-debdeb) ...
Setting up libssl1.0.0-armhf (1.0.0-1-i-debdeb) ...
pi@raspberrypi:~$ curl -LO https://github.com/ibm-messaging/iot-raspberrypi/rel
eases/download/1.0.2.1/iot_1.0-2_armhf.deb
% Total % Received % Xferd Average Speed Time Time Time Current
Load Upload Total Spent Left Speed
100 164 0 164 0 0 157 0 --:--:-- 0:00:01 --:--:-- 157
100 600 0 600 0 0 457 0 --:--:-- 0:00:01 --:--:-- 457
100 119 100 119 0 0 29217 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 (227-25+deb8u1) ...
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

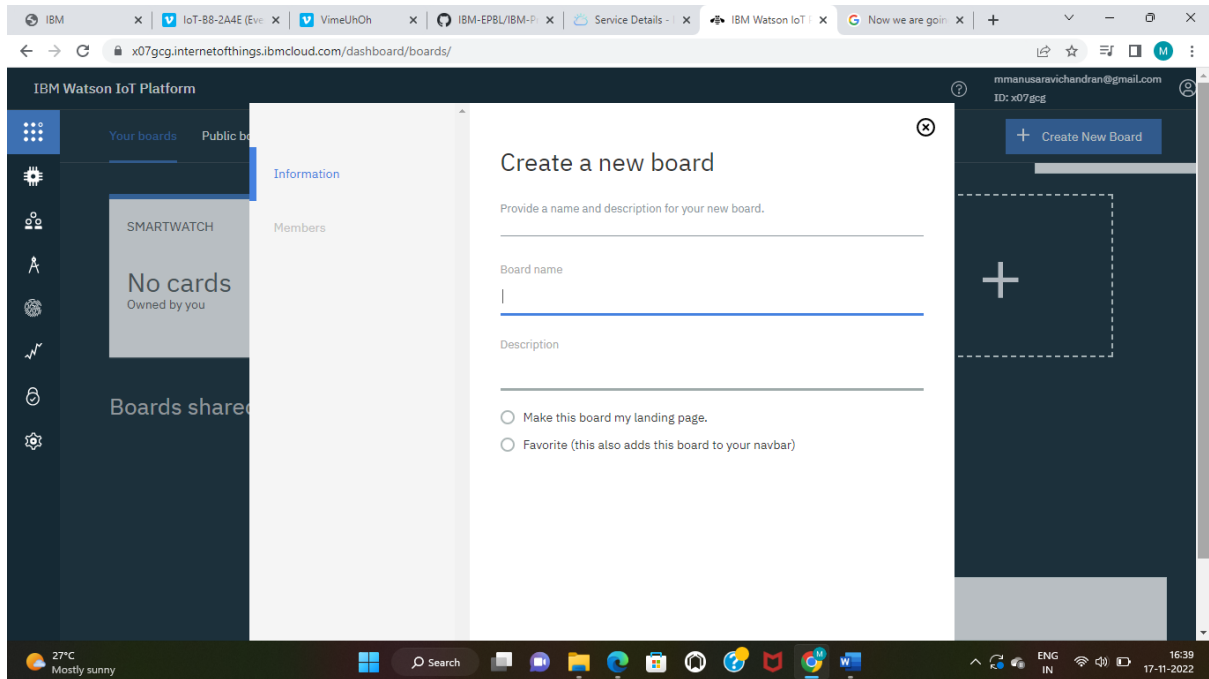
Oct 23 06:56:24 raspberrypi systemd[1]: Starting LSB: IoT service...
Oct 23 06:56:24 raspberrypi iot[2562]: Starting the iot program
Oct 23 06:56:25 raspberrypi iot[2562]: *** IoT Raspberry Pi Sample has started ***
Oct 23 06:56:25 raspberrypi iot[2562]: Config file not found. Going to Quickstart mode
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`

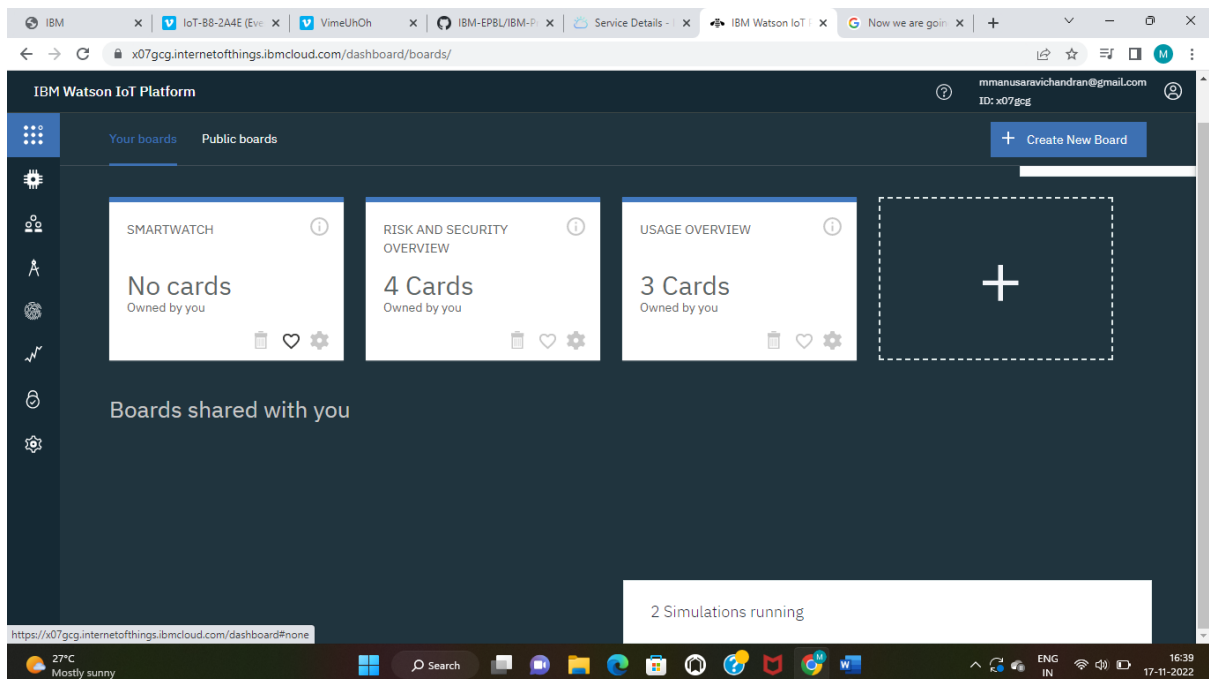
```
File Edit Tabs Help
pi@raspberrypi:~$ pip install ibmiotf
Collecting ibmiotf
  Downloading ibmiotf-0.3.0.tar.gz (50kB)
    100% |#####| 61kB 510kB/s
Collecting dicttool>=1.7.4 (from ibmiotf)
  Downloading dicttool-1.7.4.tar.gz
Collecting iso8601>=0.1.10 (from ibmiotf)
  Downloading iso8601-0.1.12-py2.py3-none-any.whl
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 1.0MB/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>=2.0.1 (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, <=2.8 (from requests>=2.5.0->ibmiotf)
  Downloading idna-2.8-py2.py3-none-any.whl (56kB)
    100% |#####| 61kB 1.7MB/s
Collecting chardet>=3.0.0, <=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, dicttool, paho-mqtt
Running setup.py bdist_wheel for ibmiotf ... done
Stored in directory: /home/pi/.cache/pip/wheels/7c/f9/45/bbc33ad957e82f7b71ba80e316d65a83d8d735e8d12e0c0418
Running setup.py bdist_wheel for dicttool ... done
Stored in directory: /home/pi/.cache/pip/wheels/45/62/59/96910b33ec6a7b2ae66a13765401b50def5468024078e12ccc
Running setup.py bdist_wheel for paho-mqtt ... done
Stored in directory: /home/pi/.cache/pip/wheels/72/d8/0d/acdc8f289011b7e7de71deebf0642fb83be0313dfff0493
Successfully built ibmiotf dicttool paho-mqtt
Installing collected packages: dicttool, iso8601, paho-mqtt, pytz, urllib3, idna, chardet, certifi, requests, requests-toolbelt, xmltodict, ibmiotf
Successfully installed certifi-2017.7.27.1 chardet-3.0.4 dicttool-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 xmltod
pi@raspberrypi:~$
```

Step-3: Checking your data sent on IBM Bluemix

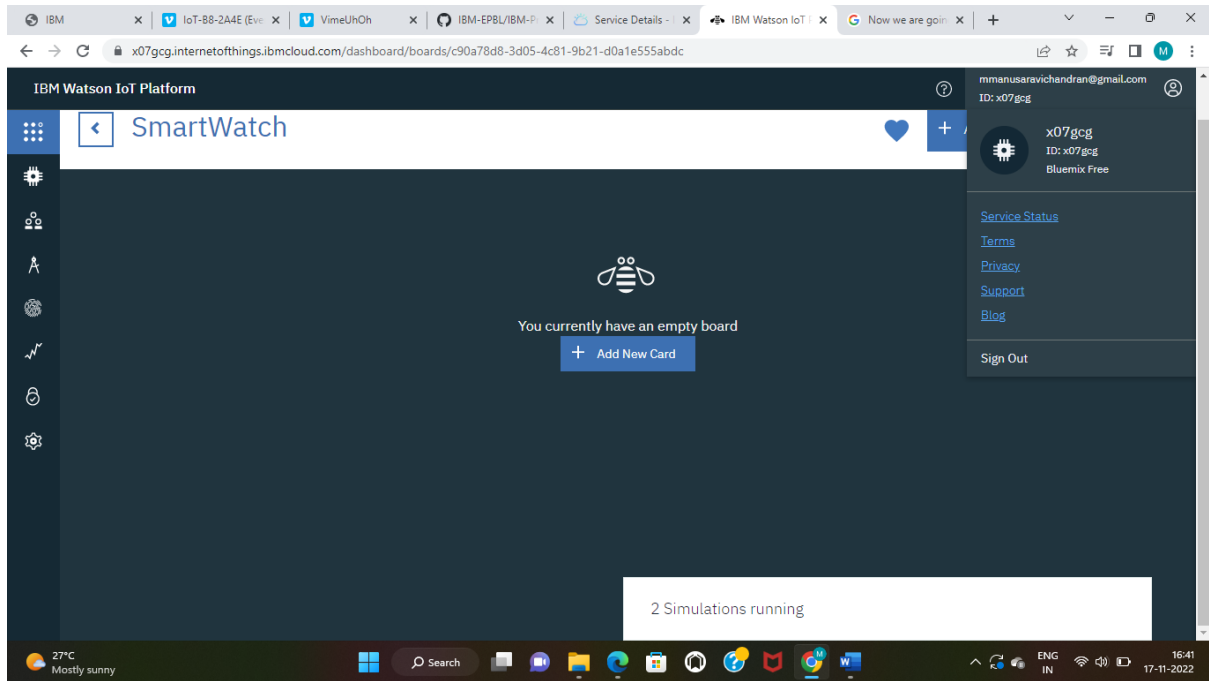
After you have sent your sensors data you can check whether it is received at your iot platform :Just look at the image below and if u see the same wifi kind of symbol on your created device then your data is being received.



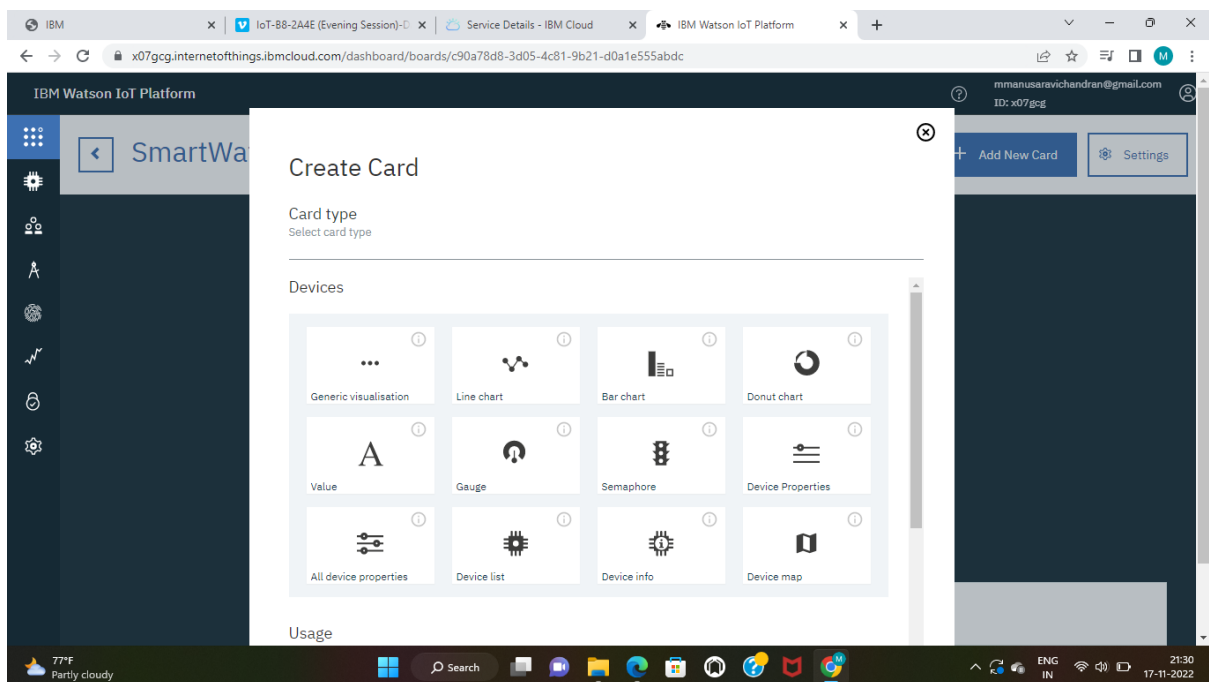
Step-4: Creating boards and cards for visualization of data:



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

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

