**Dr Vithya Ganesan**

**Professor, CSE**

**Koneru Lakshmiah Educational Foundation**

**AP**

**Problem Statement**: Design and Develop an IoT System Prototype to sense a particular parameter using sensor and send it to cloud for further processing and controlling. If you don’t want to take the data to cloud, you should monitor/control these parameters using an Android Device. You can also take the data to cloud and monitor/control through an Android device.

**1. Software modules with source code developed by the student.**

**Answer:**

**For IoT system we have developed source code program using PYTHON for real time temperature monitoring & humidity monitoring**

We were not having microcontrollers & sensors due to COVID 19 situation, so we had developed IoT system on IBM Watson cloud by simulating real time temperature & humidity sensors.

**By using hardware devices (DHT22 sensor/Raspberry pi/bread board/jumper wires/ resistors/ python), we will follow below steps for configuring IoT system**

Step 1: Python script to see how to stream real-time temperature readings collected by the DHT22. In order to run PubNub on the Pi, we used following commands on terminal:

#### Installing PubNub

**Install Python:** pi@raspberrypi ~$ sudo apt-get install python-dev

**Install pip:** pi@raspberrypi ~$ sudo apt-get install python-pip

**install PubNub:** pi@raspberrypi ~$ sudo pip install pubnub

We need to use Adafruits DHT library to be able to read the temperature values from the sensor.

The Python code to work with Adafruit's DHT sensors

**Downloading the Adafruit DHT library:**

pi@raspberrypi ~$ git clone https://github.com/adafruit/Adafruit\_Python\_DHT.git pi@raspberrypi ~$ cd Adafruit\_Python\_DHT

**Installing the library**

pi@raspberrypi ~$ sudo python setup.py install

This compiles the code for the library and then we installed it on device so any Python program can access the Adafruit\_DHT python module.

First, we import the libraries required for this project. We then initialize a PubNub object and use the publish subscribe keys which we got while signing up.

import os

import time

import sys

from pubnub import Pubnub

import Adafruit\_DHT as dht

pubnub = Pubnub (publish\_key='demo', subscribe\_key='demo')

Using the read.retry method from the Adafruit\_DHT library, we can obtain the temperature denoted by 't' and 'h' respectively.

We publish the temperature on a channel called **temp\_eon** and the humidity on **hum\_eon**. This whole thing repeats till the program is terminated so this way we can get constant temperature and humidity readings.

**By PubNub** we can view these readings remotely and with **Eon** we can create beautiful real time graphs in a matter of minutes. In this example, we are plotting the temperature as a line graph, and the humidity on a gauge graph.

def callback(message):

print(message)

while True:

h,t = dht. read\_retry(dht.DHT22, 4)

pubnub.publish('tempeon', {

'columns': [

['x', time.time()],

['temperature\_celcius', t]

]

})

pubnub.publish('humeon', {

'columns': [

['humidity', h]

]

})

### Real Time Graphs

**temp-line.html:**

eon.chart({

history: true,

channel: 'tempeon',

flow: true,

generate: {

bindto: '#chart',

data: {

x: 'x',

labels: false

},

axis : {

x : {

type : 'timeseries',

tick: {

format: '%H:%M:%S'

}

}

}

}

});

**hum-gauge.html:** We chose a gauge graph to represent the humidity values, and so the Eon script reflects this.

eon.chart({

channel: 'humeon',

generate: {

bindto: '#chart',

data: {

type: 'gauge',

},

gauge: {

min: 0,

max: 100

},

color: {

pattern: ['#FF0000', '#F6C600', '#60B044'],

threshold: {

values: [30, 60, 90]

}

}

}

});

**2.** A brief report on the prototype developed specifying

1. The sensor(s), board(s), cloud platform, programming language(s) chosen

2. Hardware wiring

3. Overall architecture and design

**Answer:**

We were not having microcontrollers & sensors due to COVID 19 situation, so we had developed IoT system on IBM Watson cloud by simulating real time temperature & humidity sensors.

**1. Sensor**: Real time temperature & humidity sensors simulated on IBM Watson cloud

Board: Raspberry Pi

Cloud Platform: IBM Watson cloud

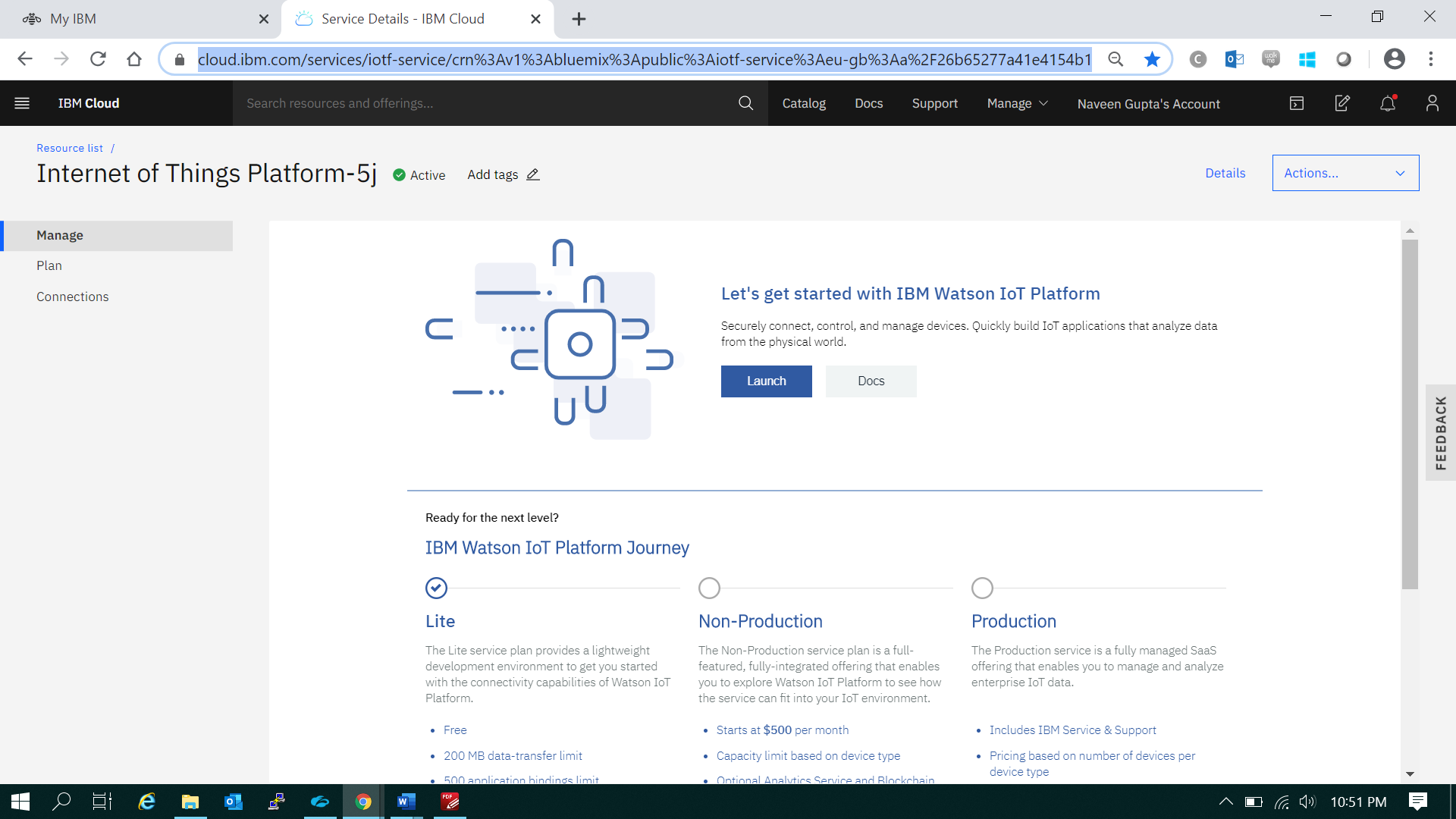
Programming language: Python

Steps involved for creating IoT system on **IBM Watson cloud**

1. Create an IoT service

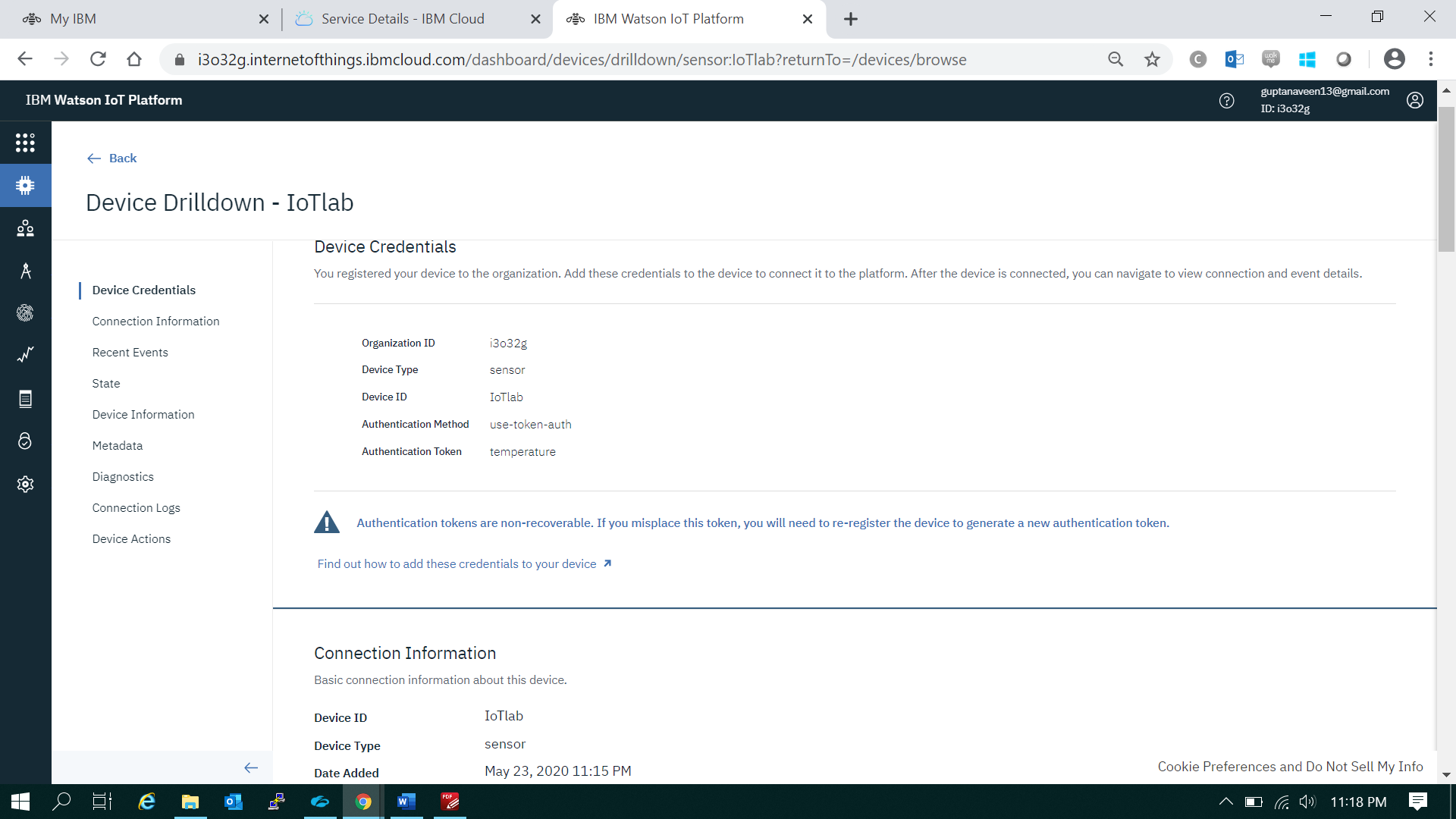
2. Register a device to your organization

3. Send and view sensor data

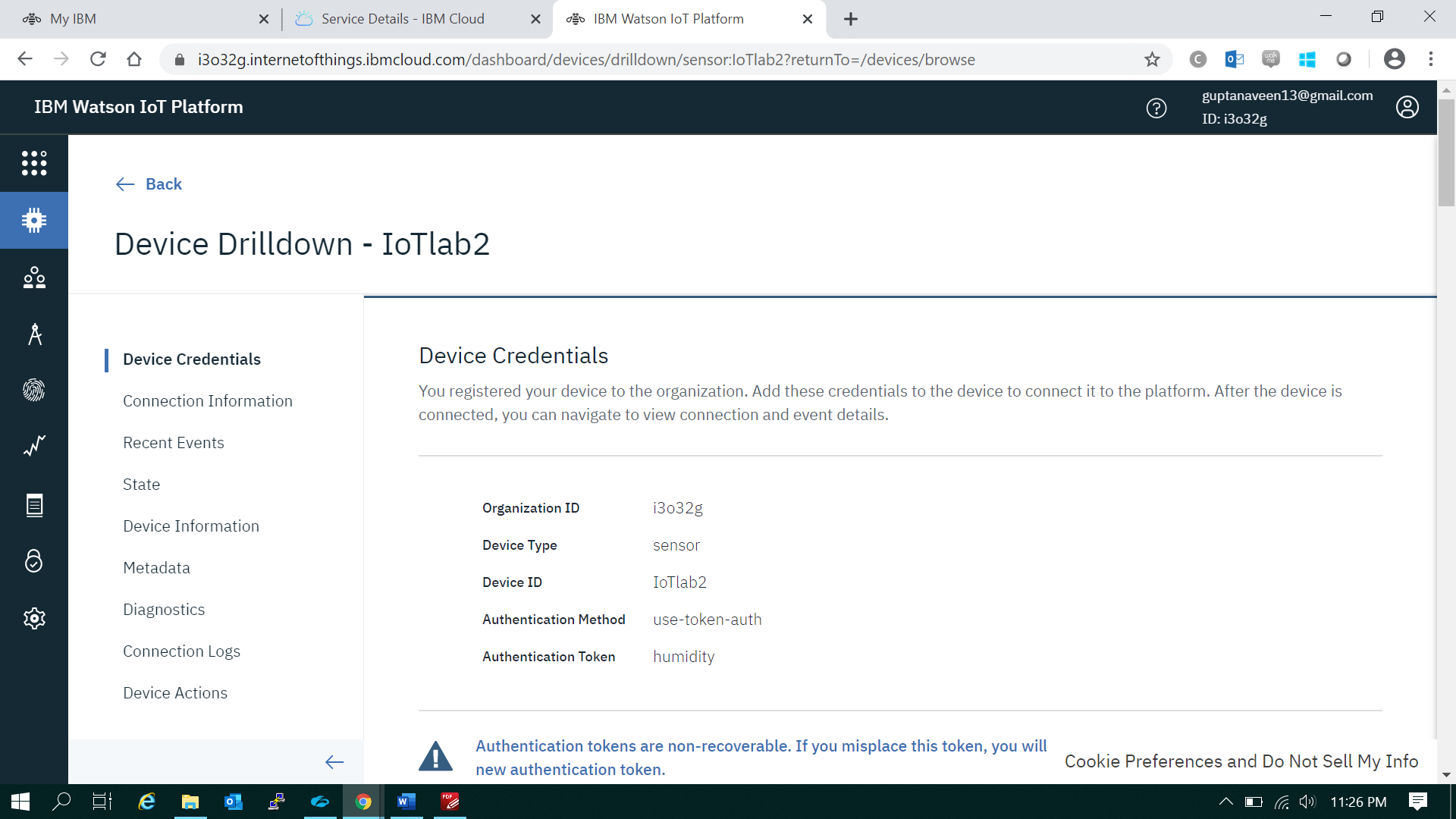


**Launch & then in the upper right click “+ Add Device”**

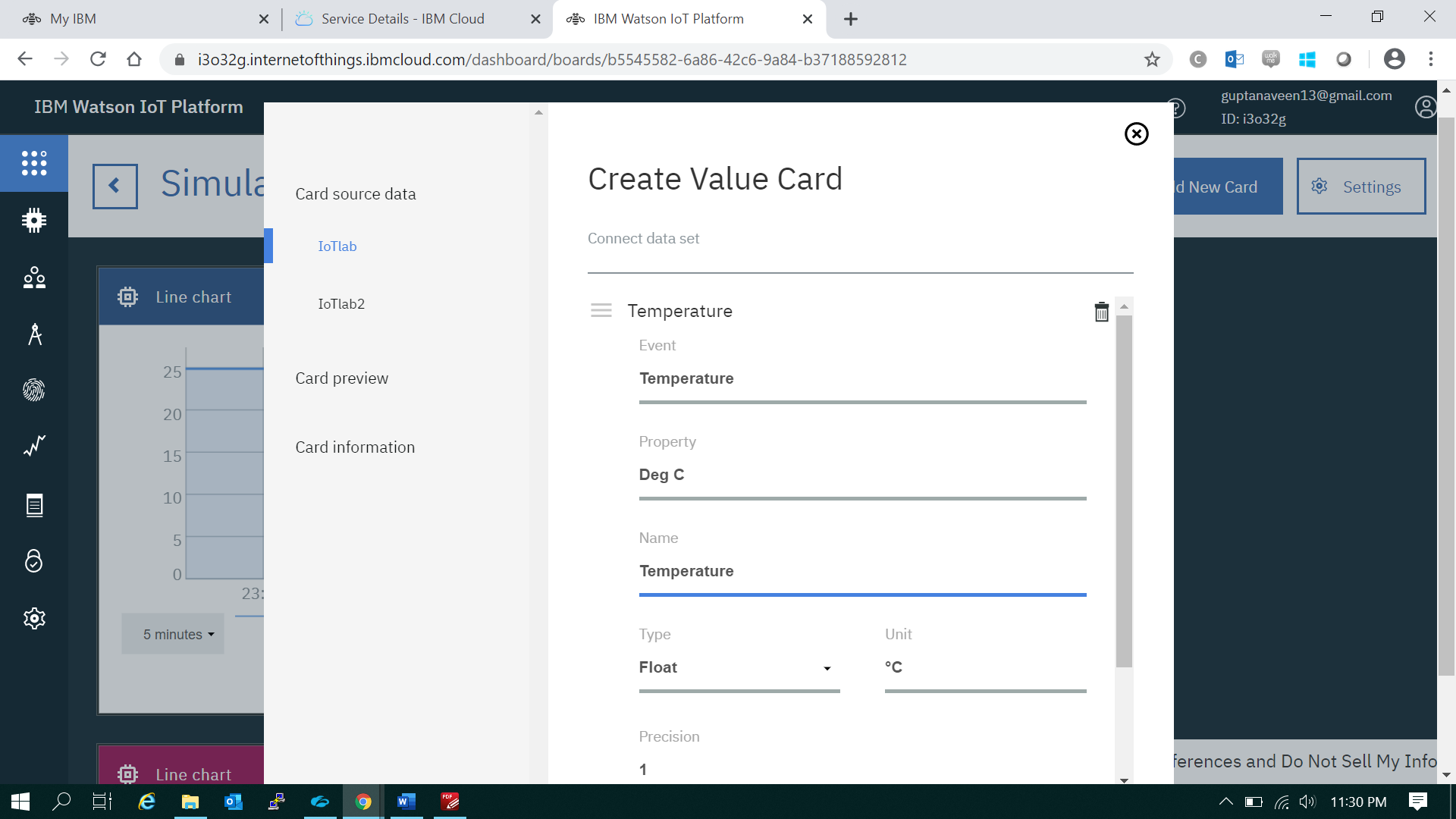
**Create device 1: Temperature Sensor & once device is created device credentials will appear like below**



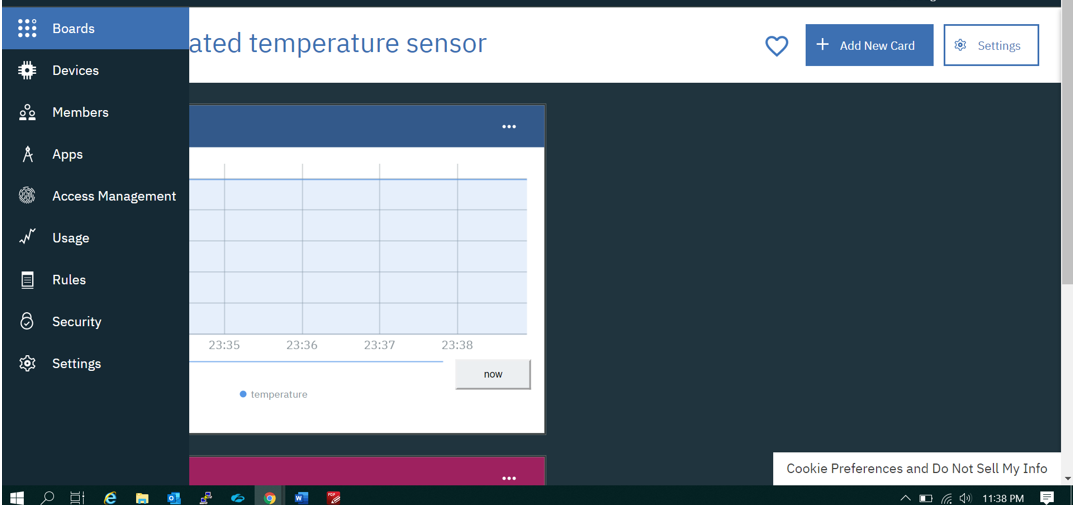
**Similarly create device 2: Humidity Sensor**



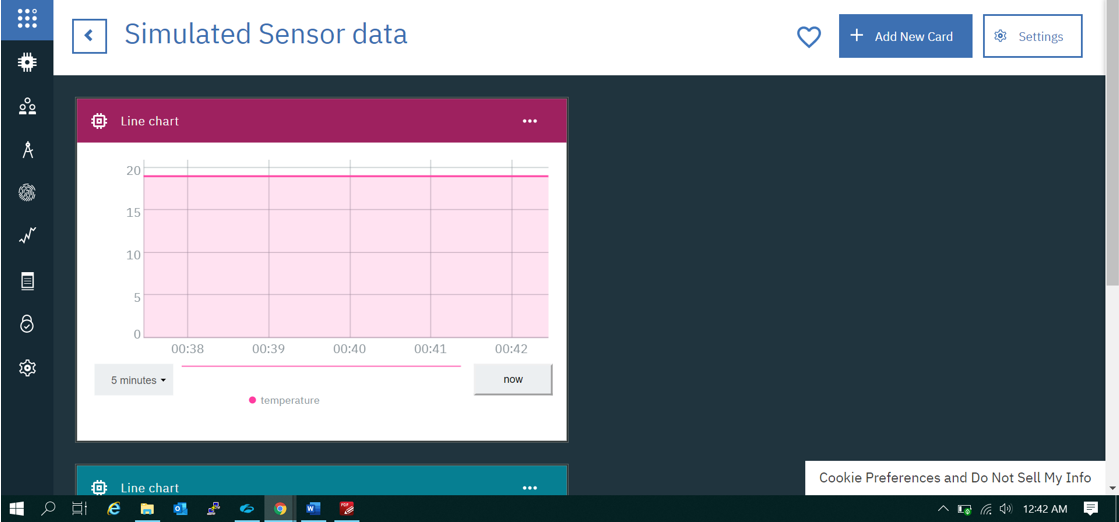
By clicking dashboard create value card for temperature & humidity sensors

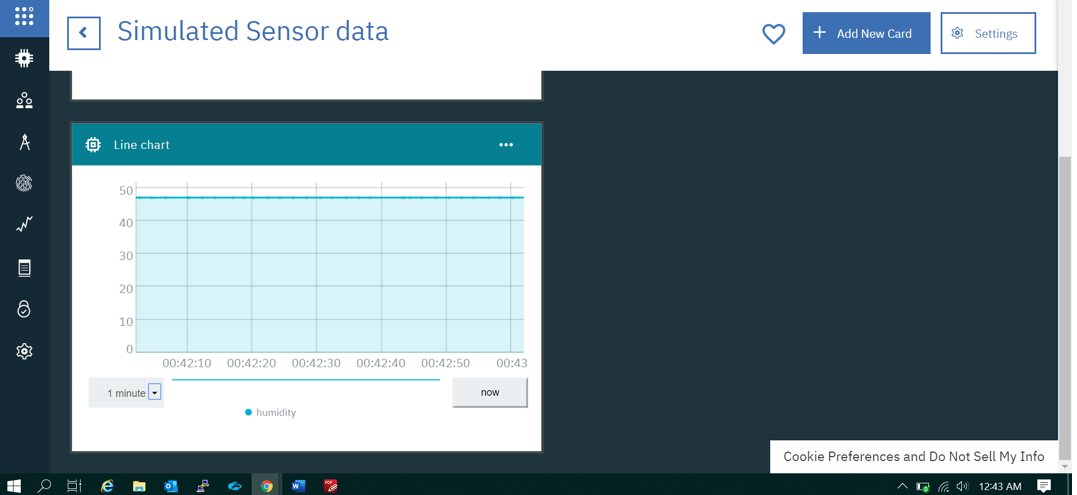


Go to dashboard once connect data set is entered



**Below cards are created for simulated sensor data for monitoring real time temperature & humidity**

****

****

#### Send Sensor Data to IBM Watson cloud

1. Open the following link in a new tab: [http://watson-iot-sensor-simulator.mybluemix.net/](http://watson-iot-sensor-simulator.mybluemix.net/?cm_sp=dw-bluemix-_-recipes-_-devcenter)

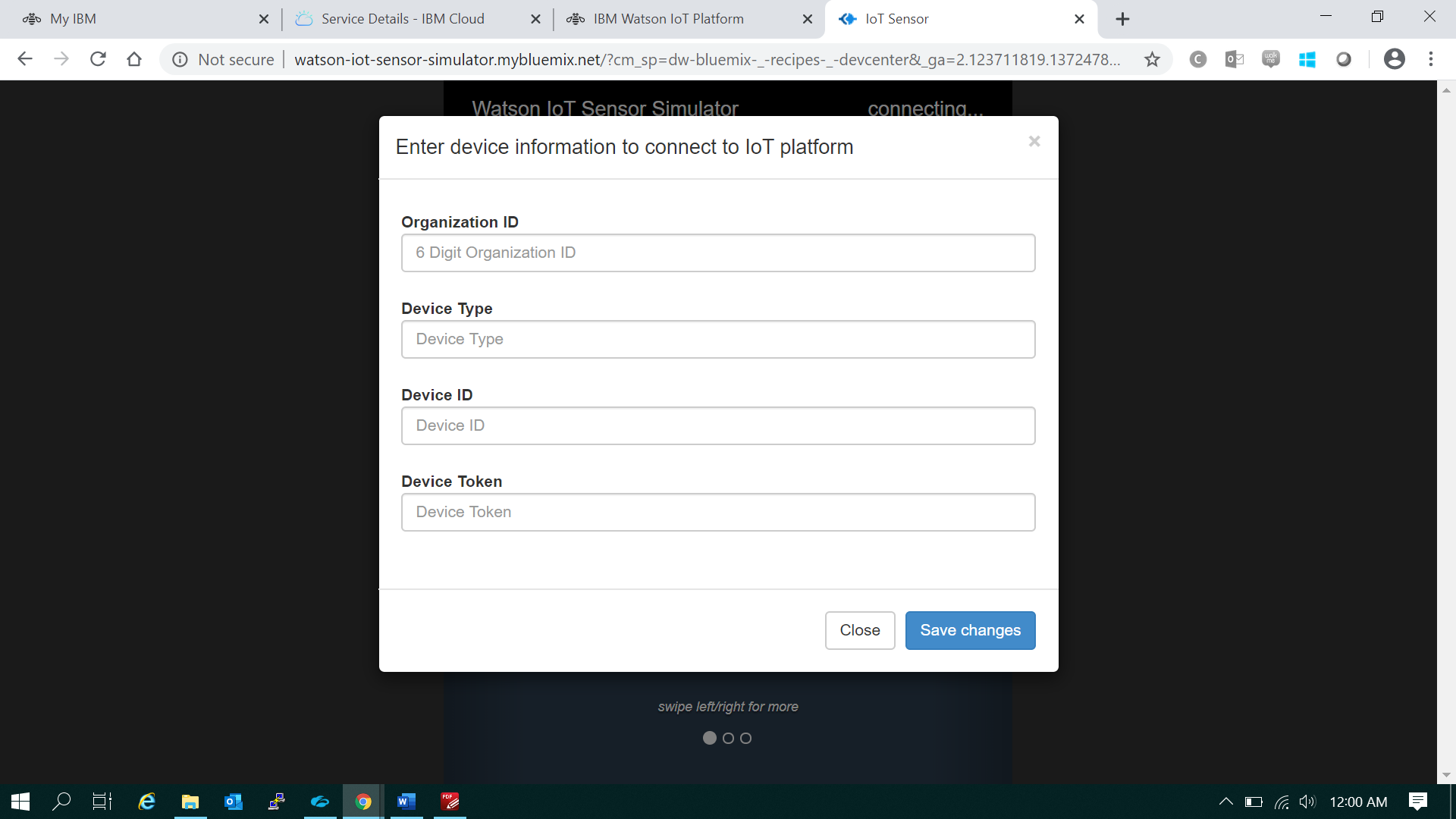
This is a simulated sensor that is capable of sending temperature, humidity, and object temperature data.

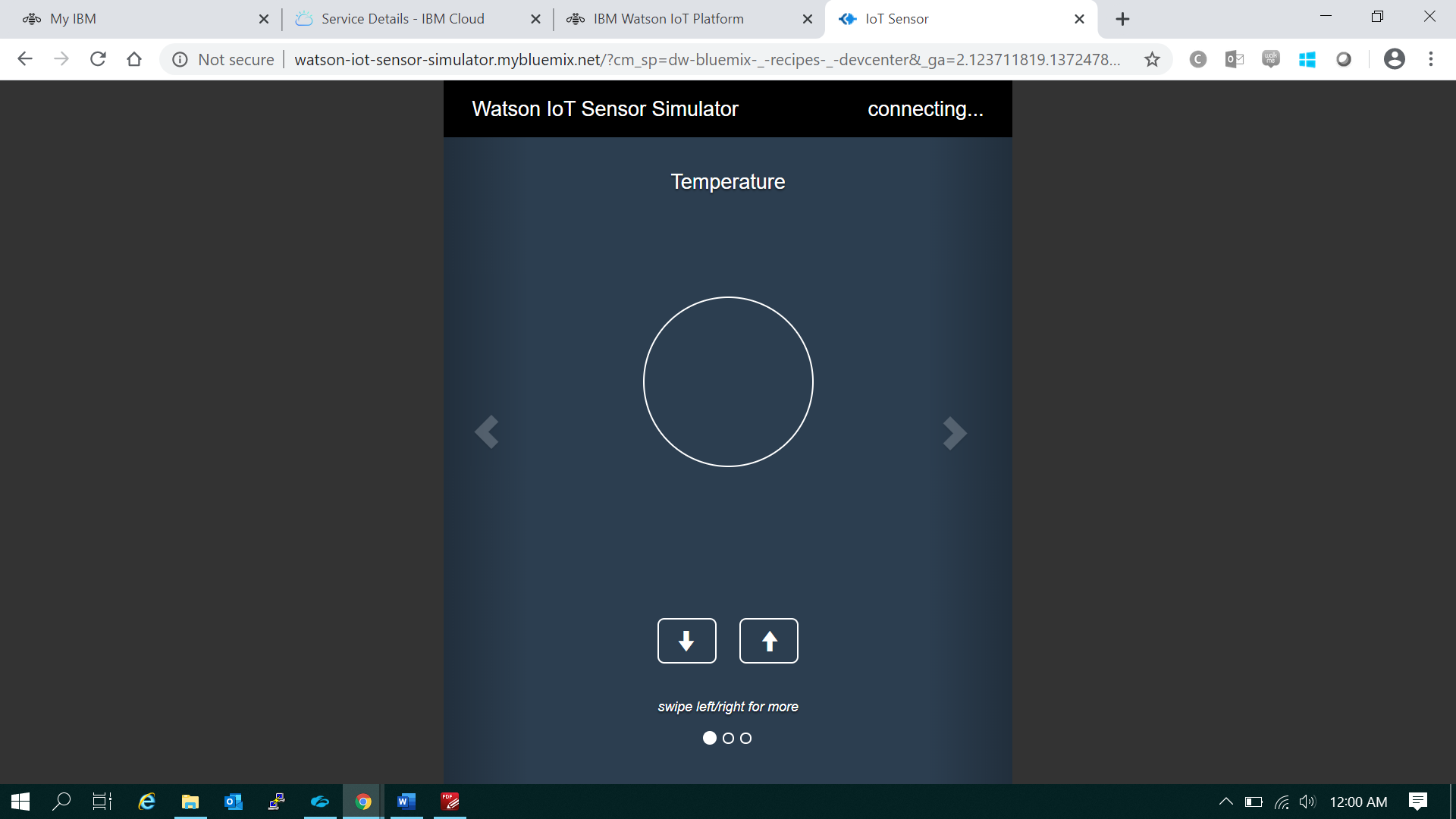
The value being sent can be changed by interacting with the web UI of the simulator.

This sensor connects to the IoT Platform and sends an ‘iotsensor’ event every 2 seconds while active.

Keep a tab or window open with the sensor to continue sending data.

2. Fill in the required information. The device token is your device’s unique authentication token. Click “Save changes”

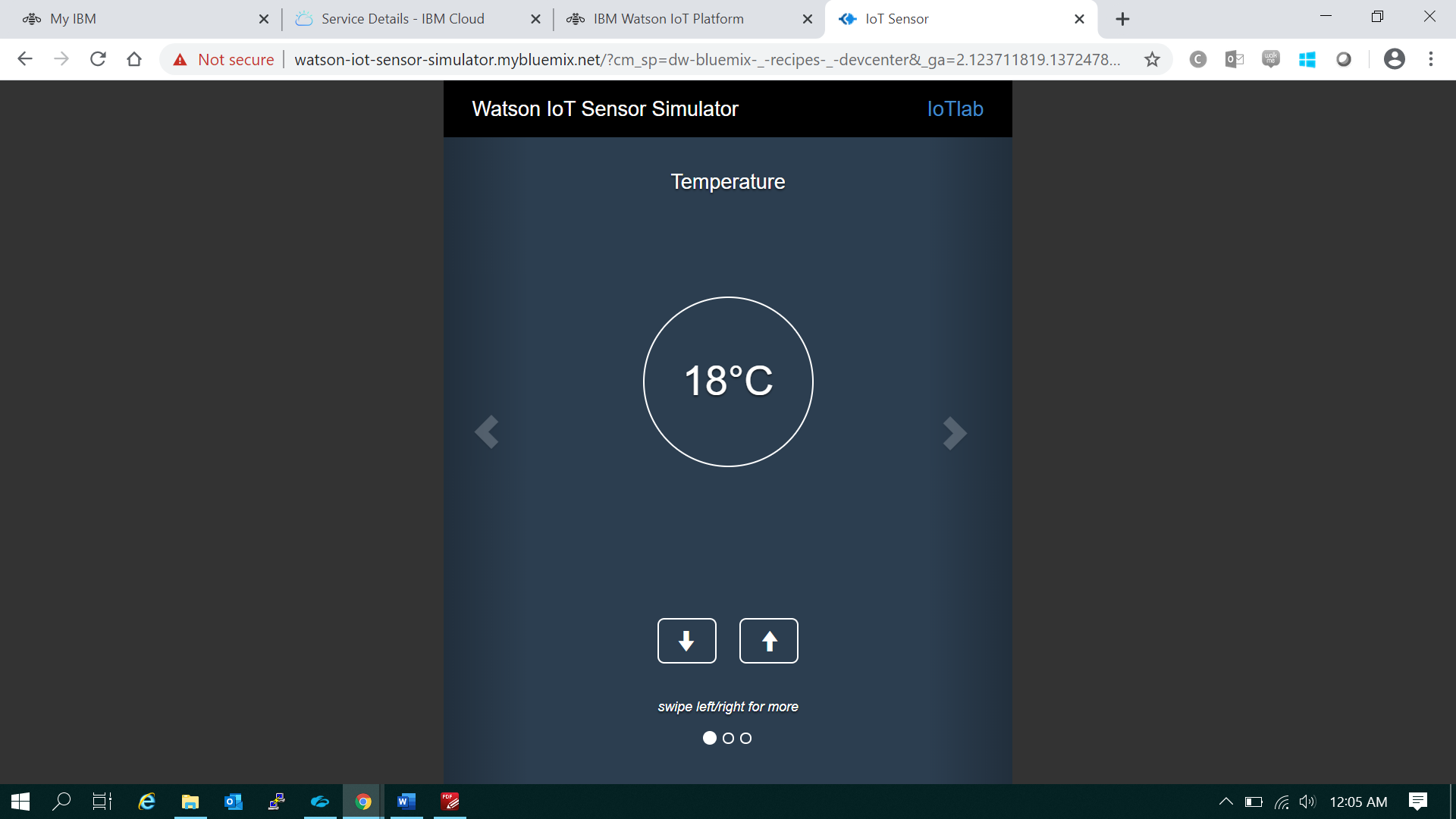




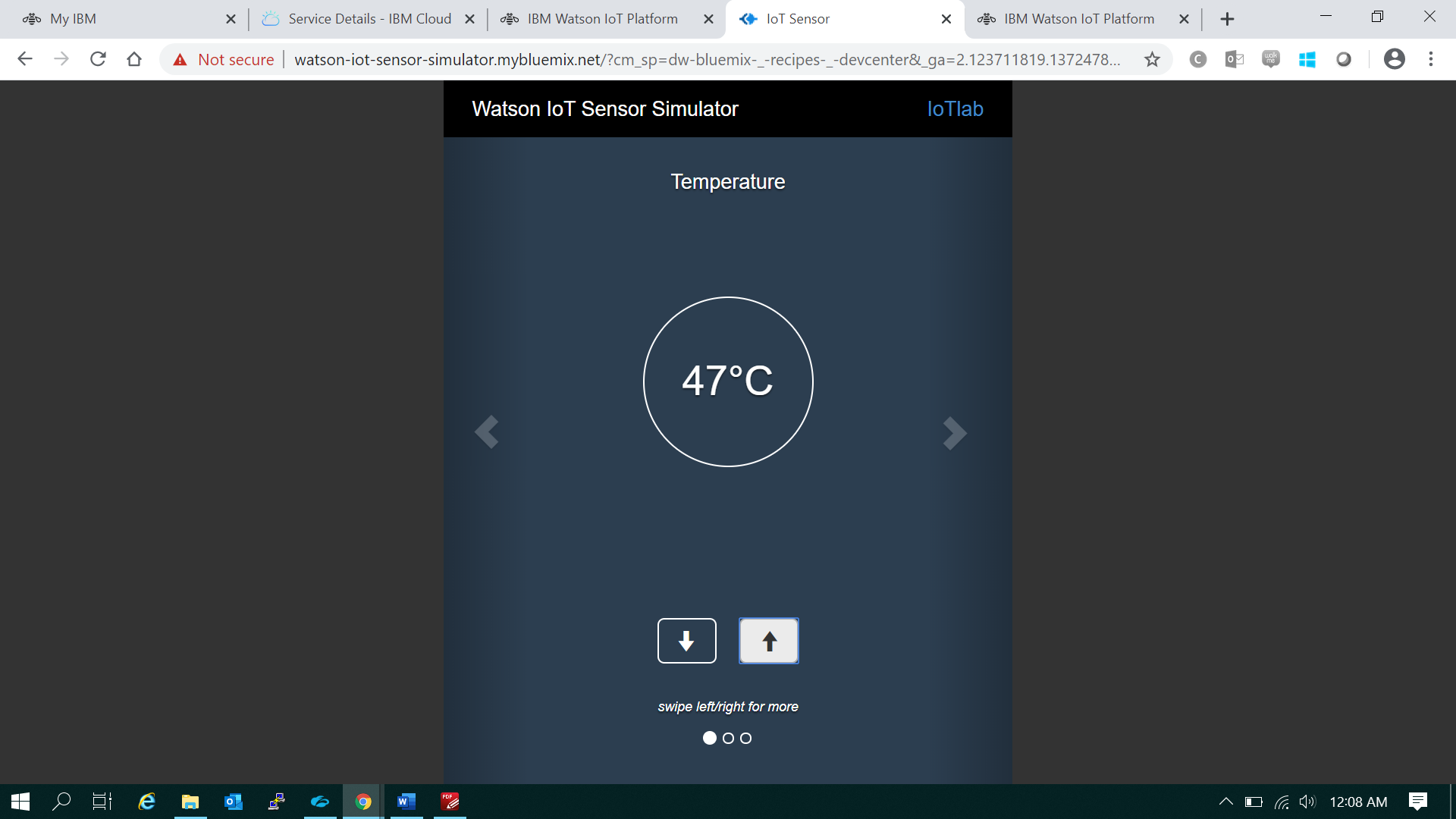
If device is connected, the “connecting…” in the upper right corner will now be replaced by device ID.

Leave this tab open to continuously send data.

Click on device ID to be brought to dashboard, otherwise refresh the page and enter information again.

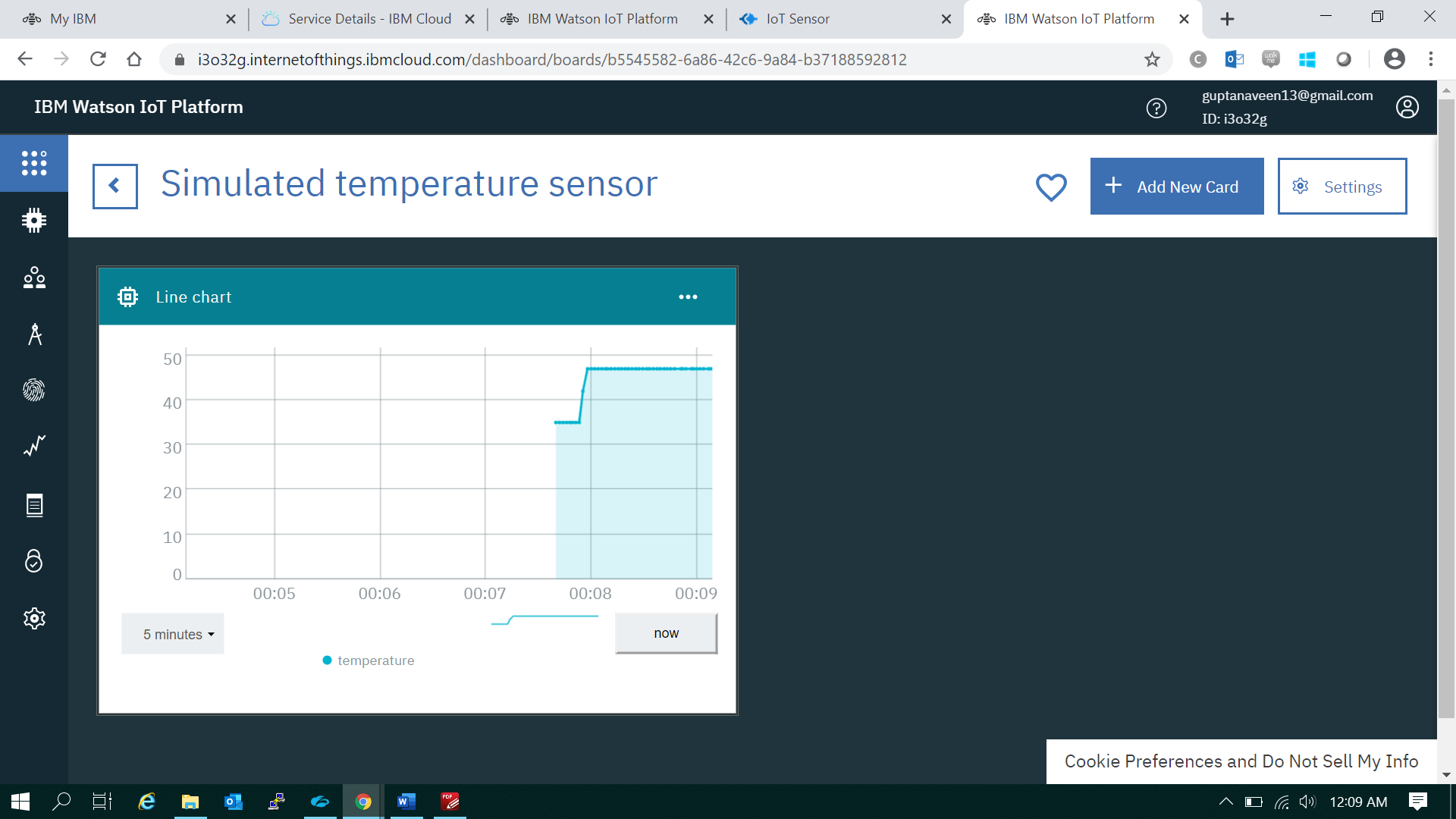


In above simulated temperature sensor, we can increase or decrease temperature value, based on variation in sensor values, line chart graph will display real time temperature on card created on IBM Watson IoT platform

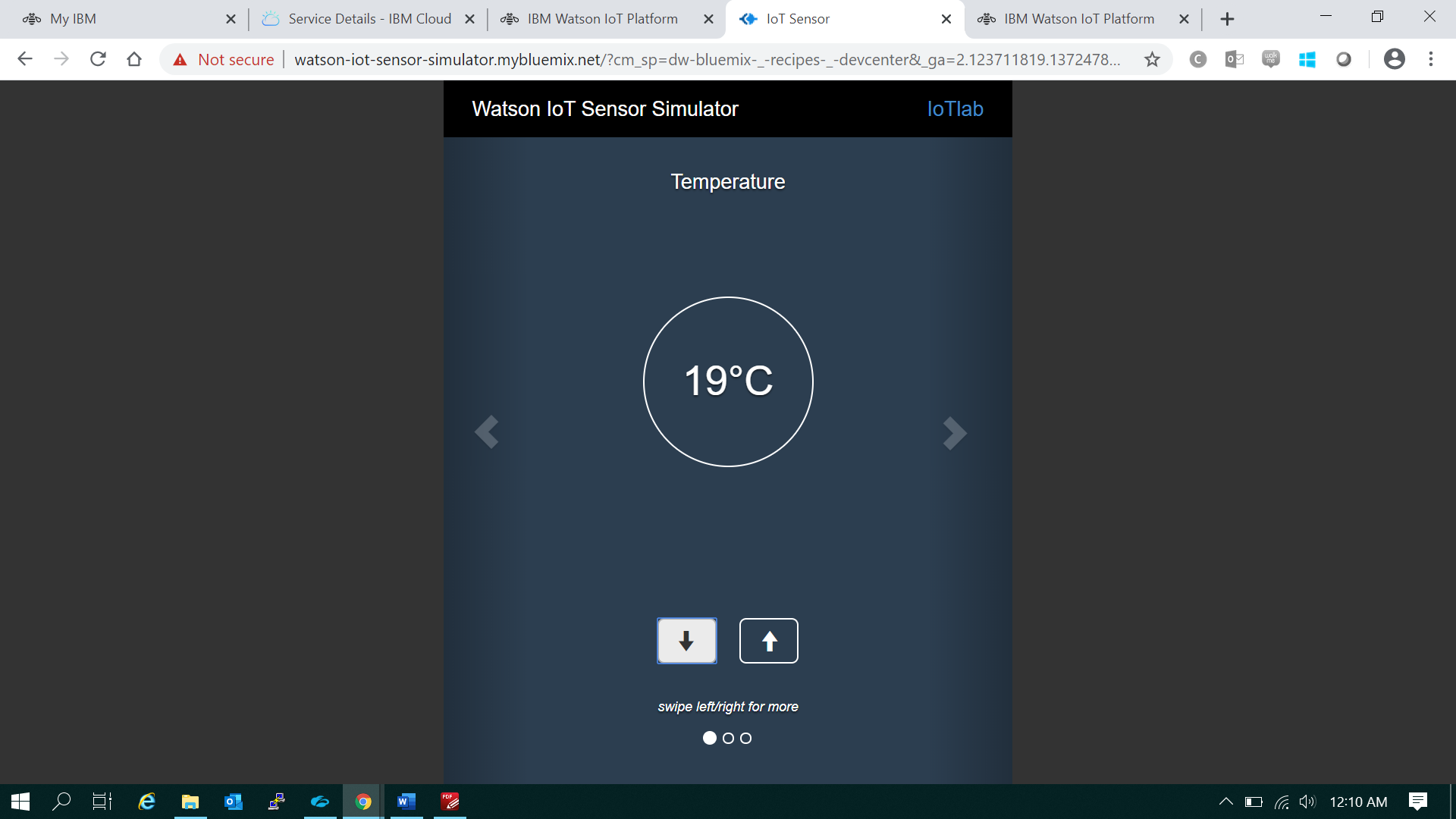


# **Visualizing Data in Watson IoT Platform**

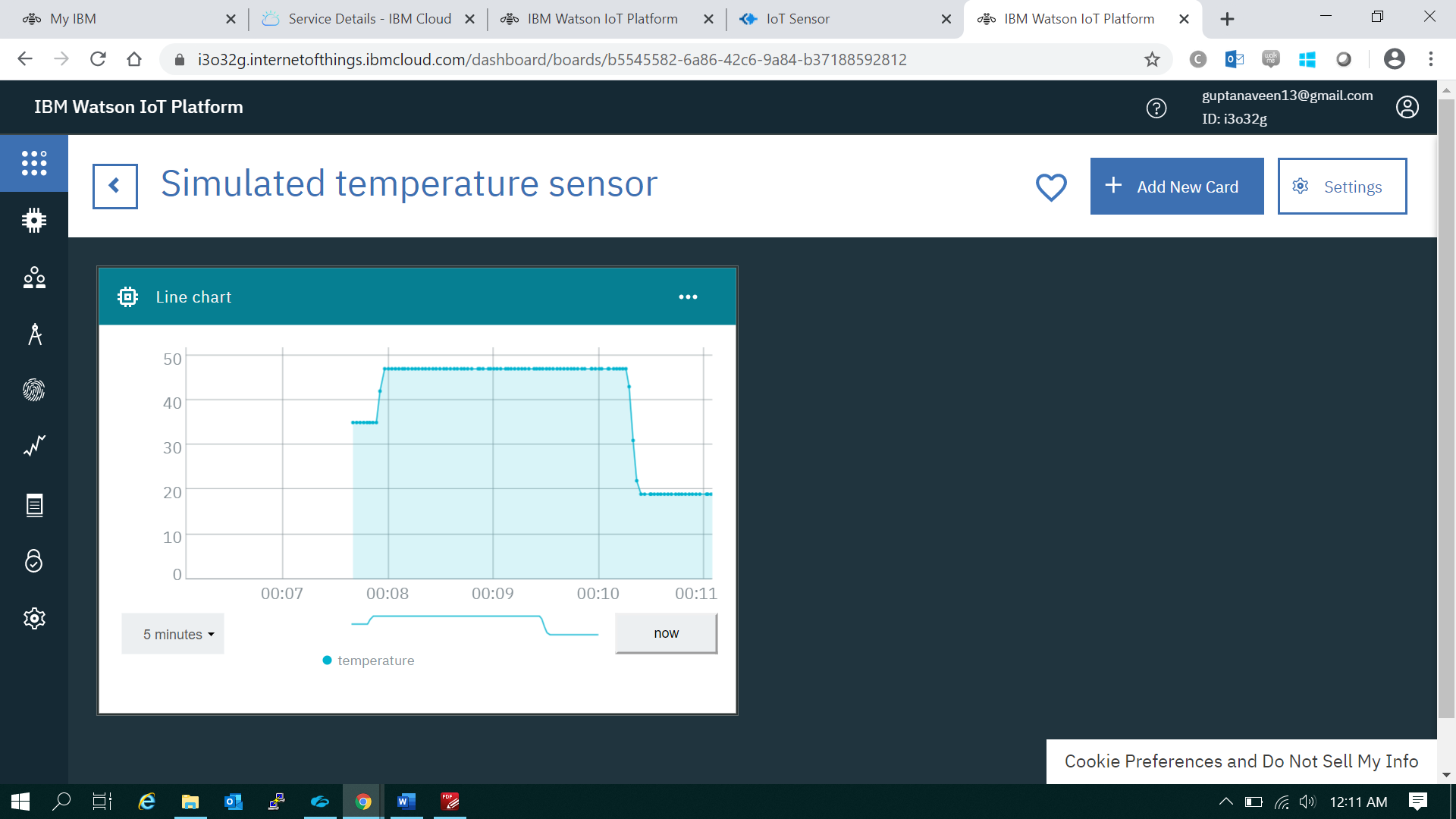
# Real time value of temperature sensor appearing on Watson IoT Platform

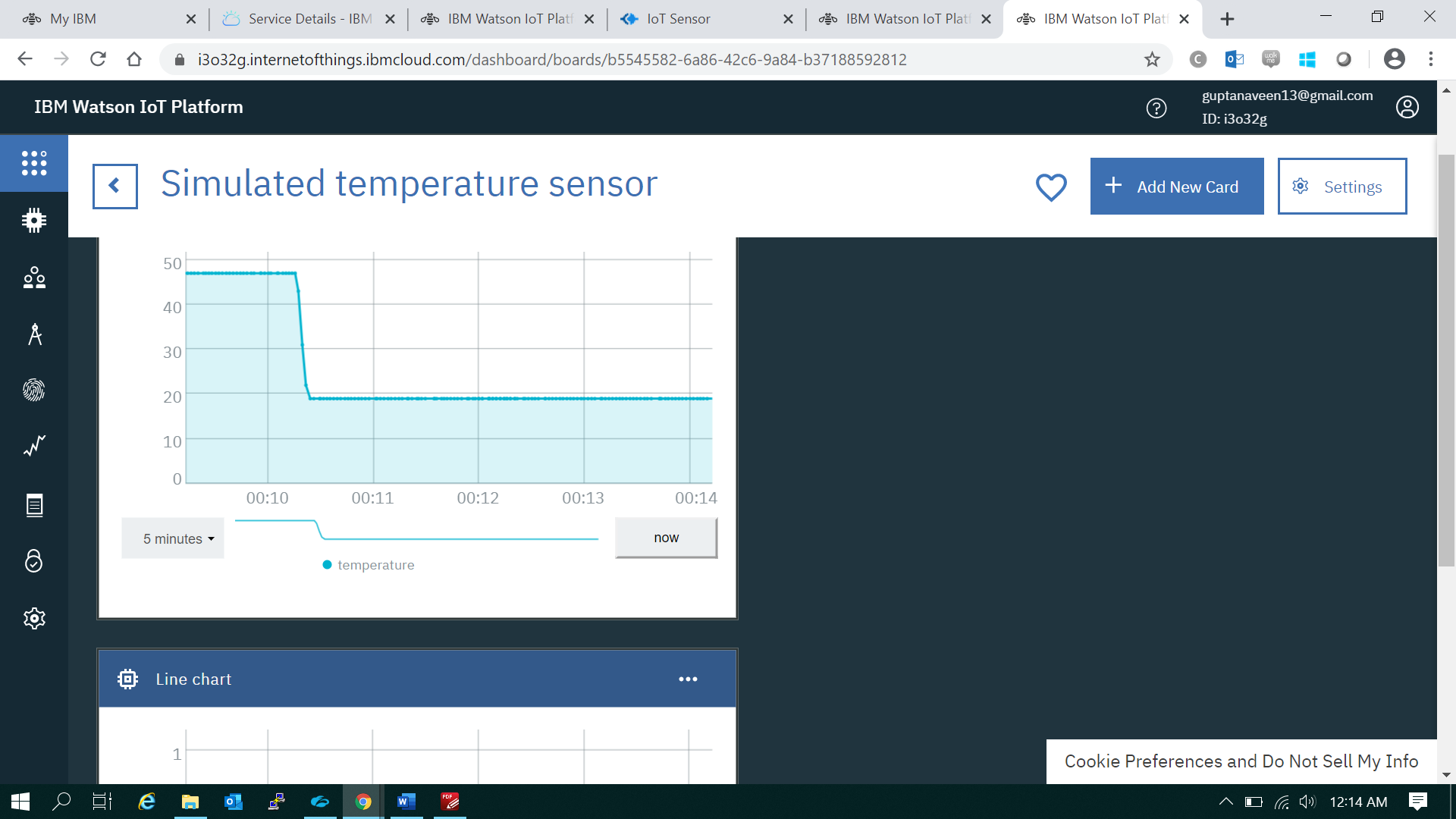


Further when we reduce temperature again to 19deg C, line chart graph will vary accordingly



Real time Temperature line chart with temperature reduced to 19deg C

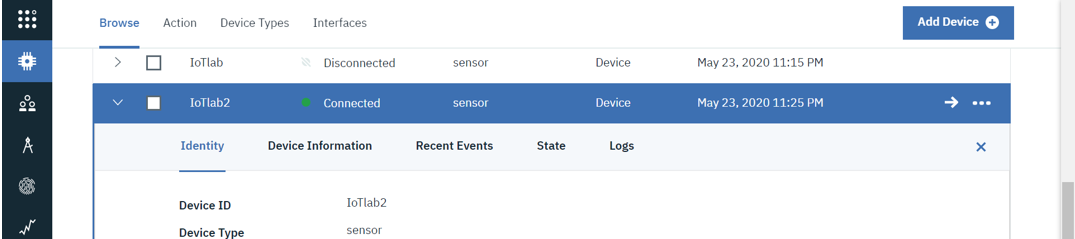




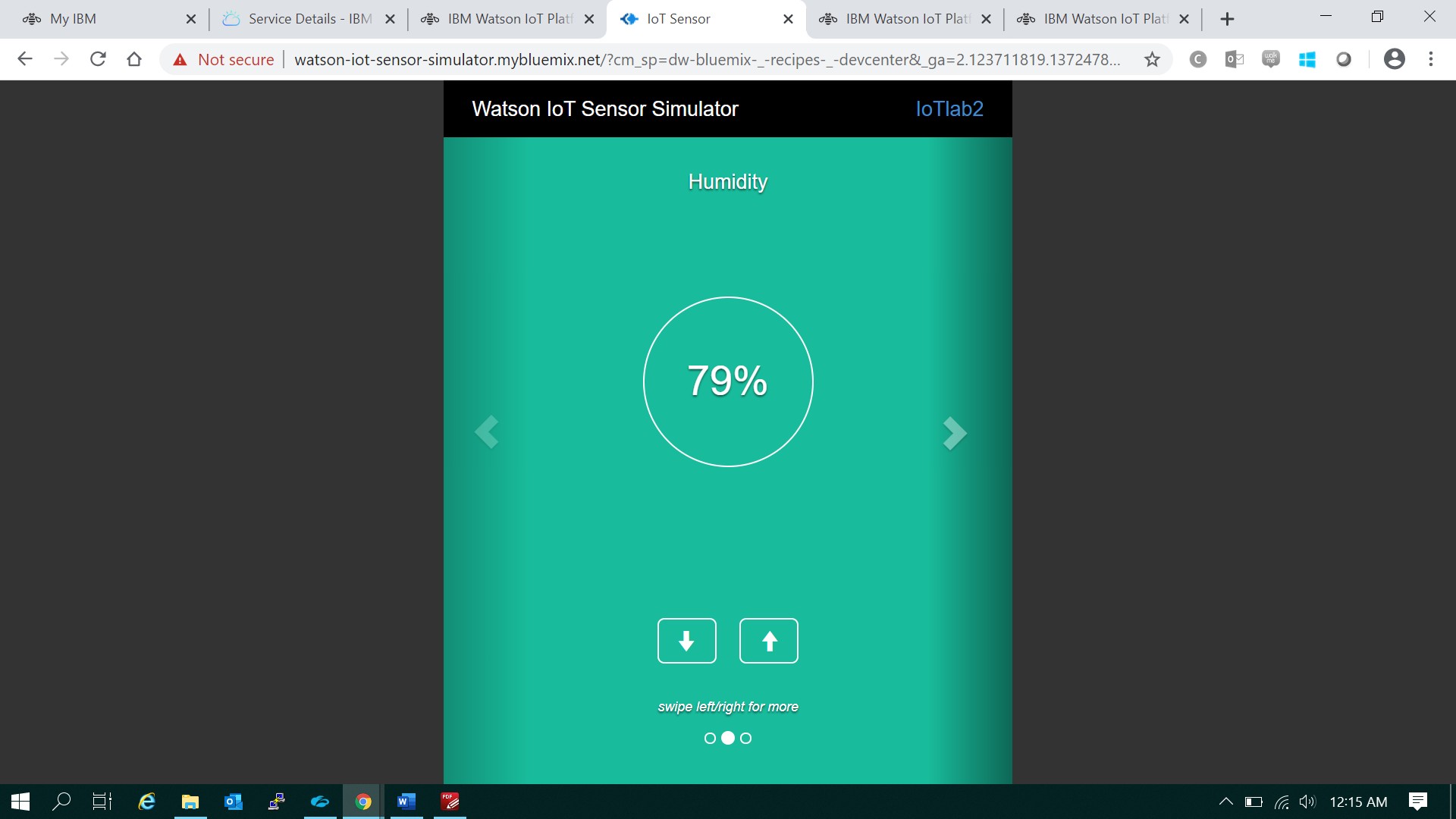
Similarly, in next step we created device 2 humidity sensor & configured with Watson simulator / platform

**Humidity sensor:**

Created one more device with device ID IoTlab2 for measuring humidity, once device is created it will appear like in below screen shot



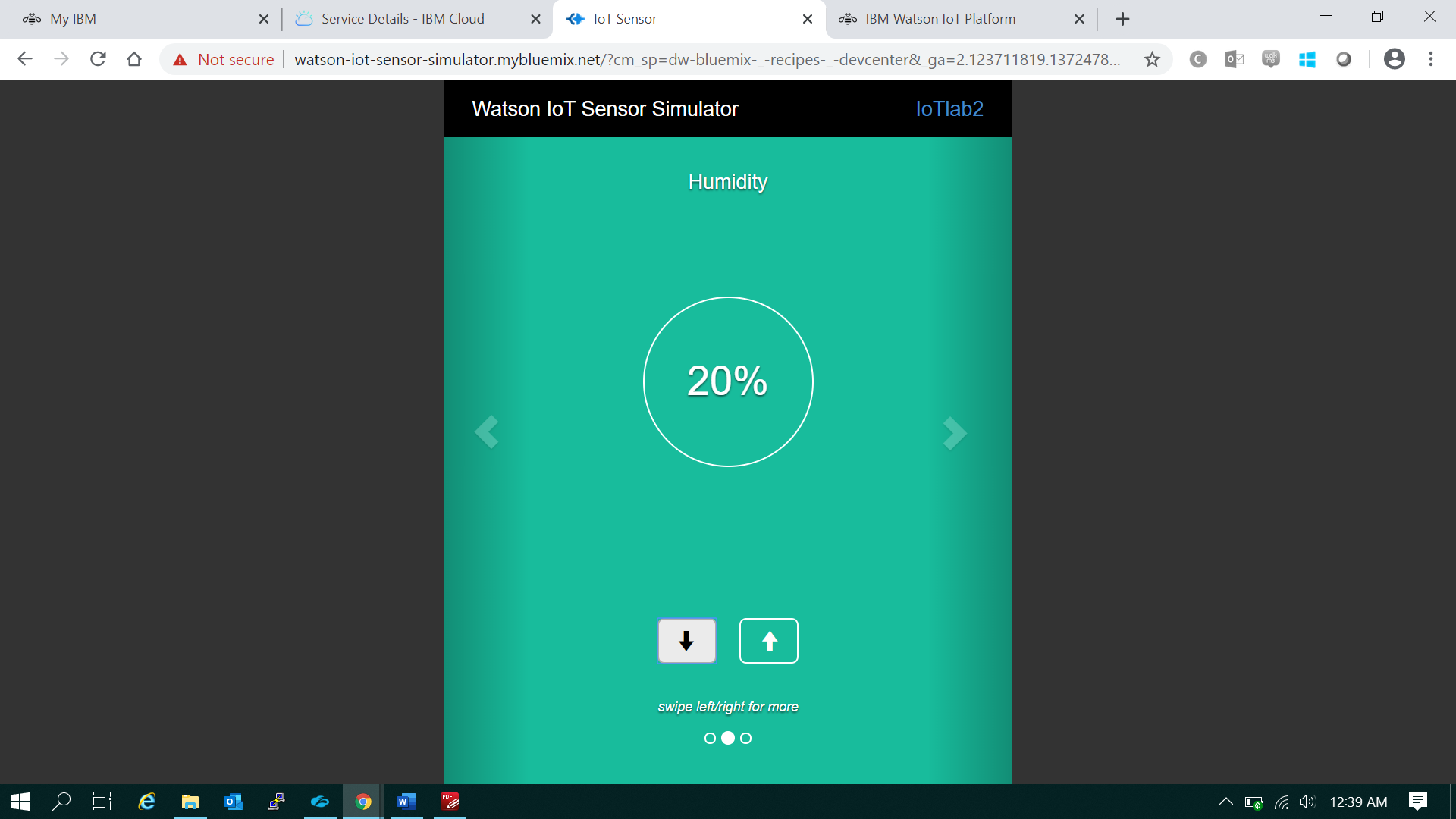
Then once device & cards are created, we need to simulate humidity sensor data on IBM Watson cloud



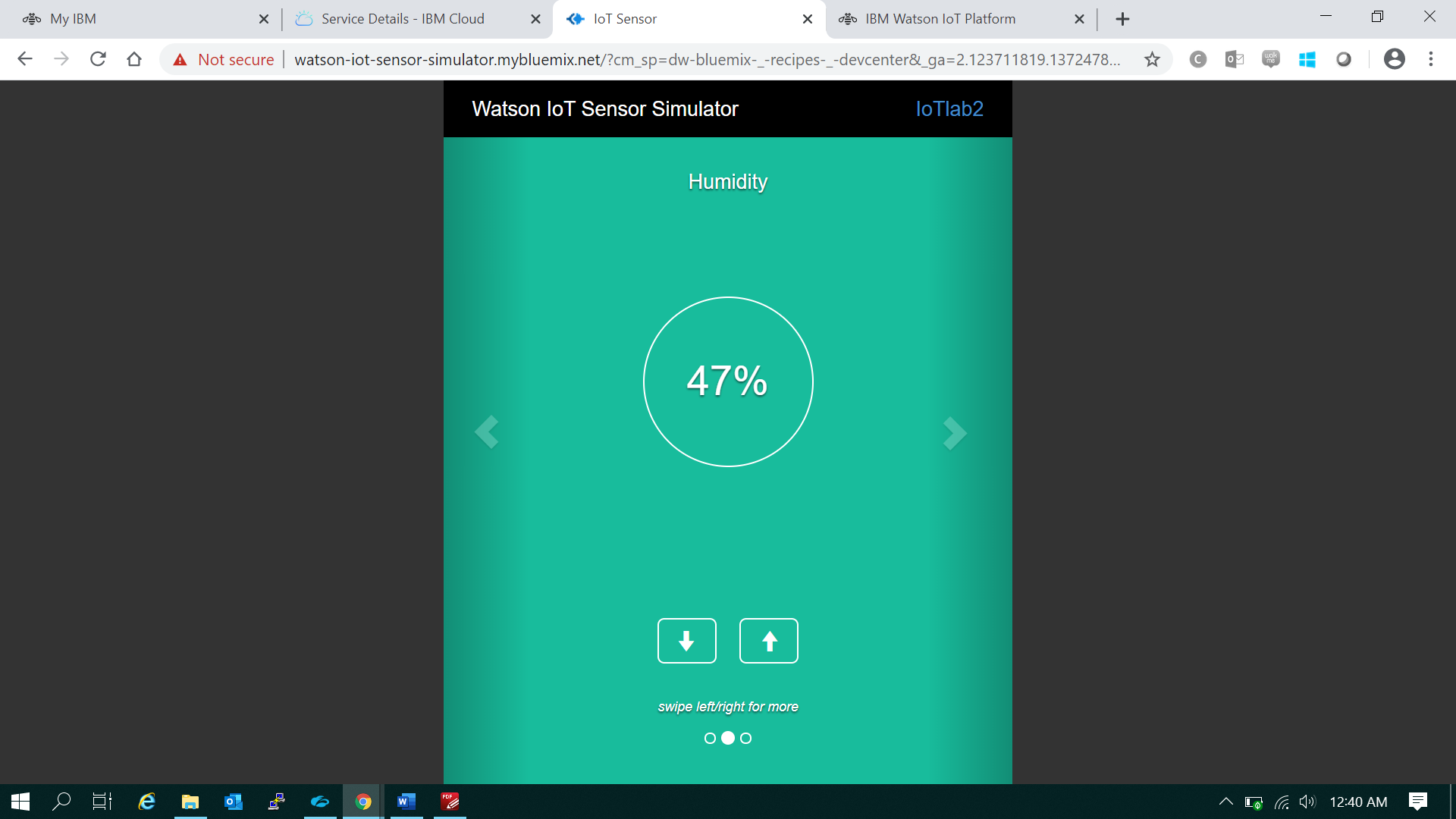
humidity real time simulated sensor data on **IBM Watson IoT platform** as per below screen shot

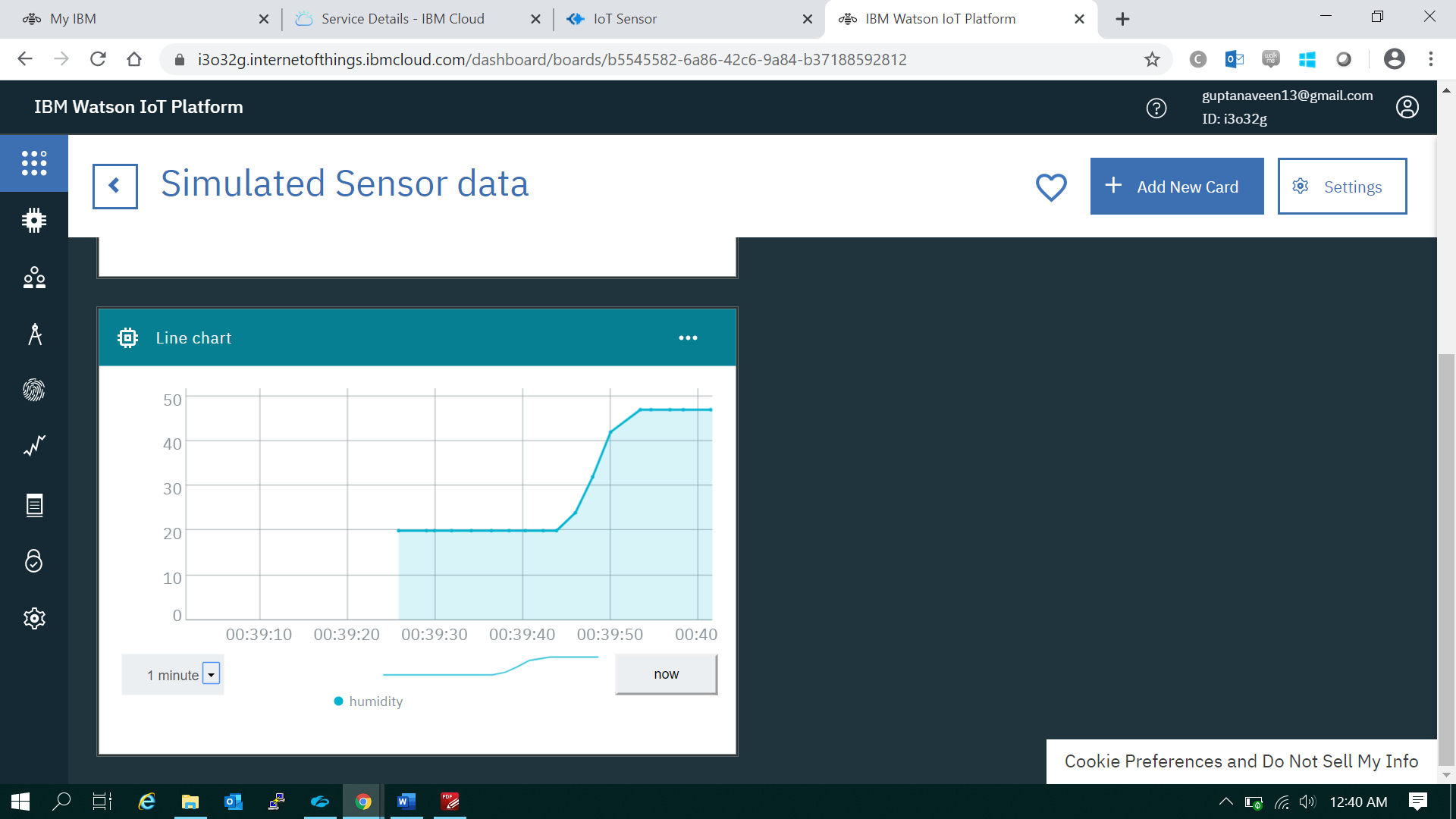


As we vary humidity value on Watson IoT sensor simulator, line chart graph varies accordingly as per below screen shots

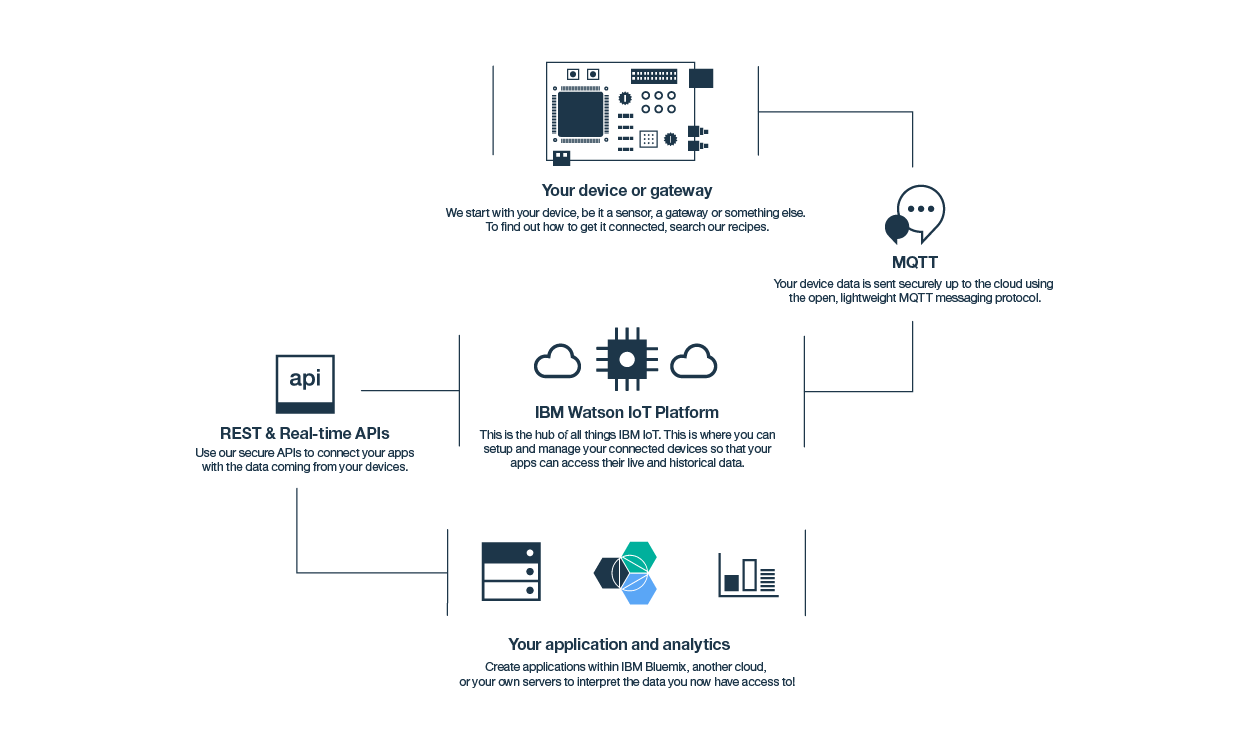


When we change humidity value from 20% to 47%, we can see real time variation in line chart of **IBM Watson IoT platform**





**Overall architecture and design**



Conclusion: Humidity of real time simulated sensor data on IBM Watson IoT platform is deployed.