SPRINT 2

Model Creation

Team ID	PNT2022TMID26635
Project Name	Project – IoT based Smart Crop Protection for Agriculture.

Model Creation:

➢ Pre-processed data are sent to the cloud in order to be processed, analysed, or modelled in order to build smart applications. Because IoT devices usually don't emit data at regular intervals, IoT time series data is highly irregular regarding their sampling rate within as well as across devices.

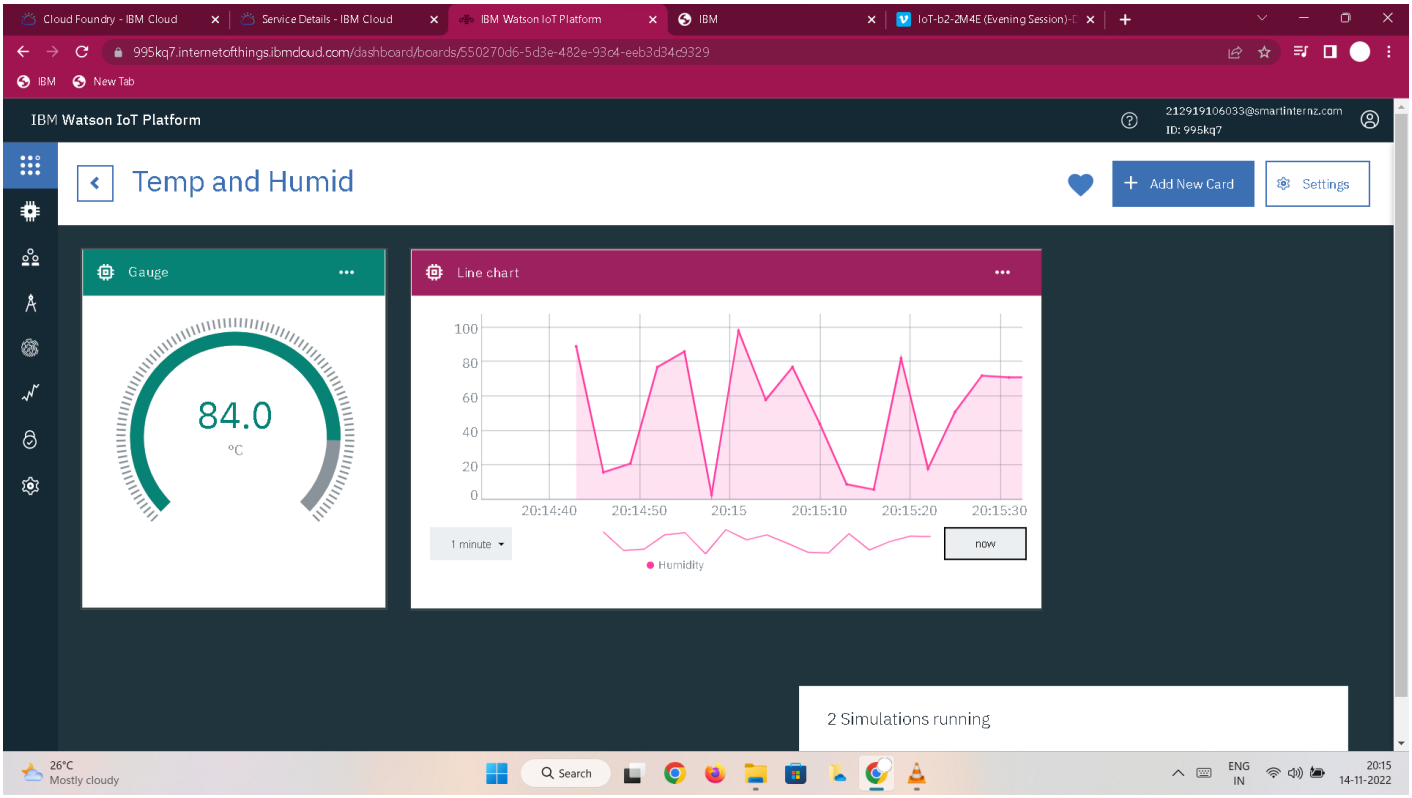
- ✓ Name: - Smart Crop Protection for Agriculture
- ✓ Device Type: Detecting the motion of animals and birds
- ✓ Kind: Sensor
- ✓ Producer: By the motion of object.
- ✓ Frequency: Every time when the motion is detected.

5.Building Project:

5.1 Connecting IoT Simulator to IBM Watson IoT Platform:

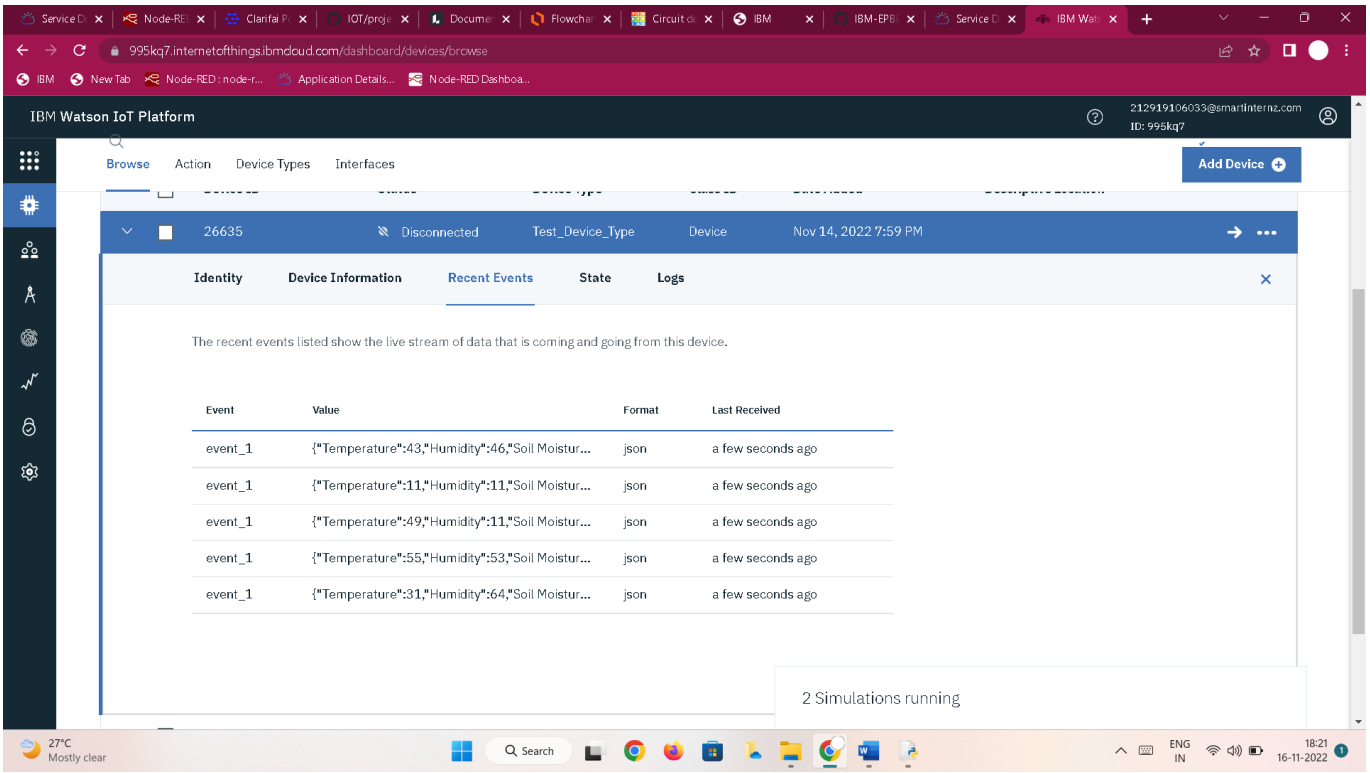
- ➢ Open link provided in above section 4.3
- ➢ Give the credentials of your device in IBM Watson IoT Platform
- ➢ Click on connect
- ➢ My credentials given to simulator are:

- ✓ Organization ID : 995kq7
- ✓ Device Type: Test_Device_Type
- ✓ Device ID : 26635
- ✓ Authentication Method : use-token-auth
- ✓ Authentication Token : o3d471A?EzrQoOU3Y_



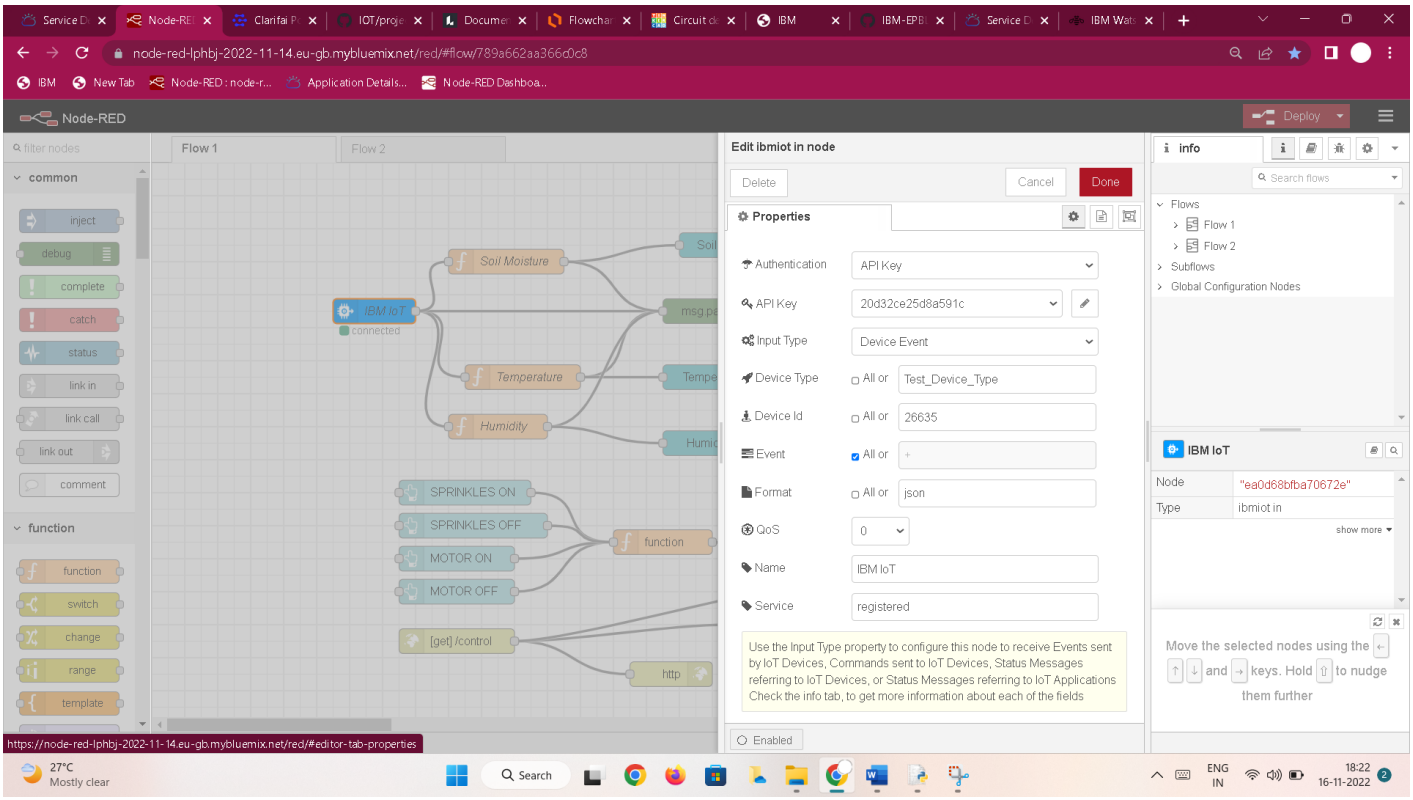
You can see the received data in graphs by creating cards in Boards tab

- ➢ You will receive the simulator data in cloud
- ➢ You can see the received data in Recent Events under your device
- ➢ Data received in this format(json)



5.2 Configuration of Node-Red to collect IBM cloud data:

The node IBM IoT App In is added to Node-Red workflow. Then the appropriate device credentials obtained earlier are entered into the node to connect and fetch device telemetry to Node-Red.



Once it is connected Node-Red receives data from the device

Display the data using debug node for verification

Connect function node and write the Java script code to get each reading separately.

The Java script code for the function node is

- ✓ Type msg.payload=msg.payload.Temperature
- ✓ Type msg.payload=msg.payload.Humidity
- ✓ Type msg.payload=msg.payload.HazardousGas
- ✓ Type msg.payload=msg.payload.d.Pressure

Finally connect Gauge nodes from dashboard to see the data in UI