

## Sprint Delivery – 2

Date	17 November 2022
Team ID	PNT2022TMID28572
Project Name	Hazardous Area Monitoring for Industrial Power plant powered by IoT
Maximum Marks	4 Marks

### Connecting IoT Simulator to IBM Watson IoT Platform:

Give the credentials of your device in IBM Watson IoT Platform. Click on connect My credentials given to simulator are:

Organization ID: ph99dh

Device Type: NodeMCU

Device ID:123456

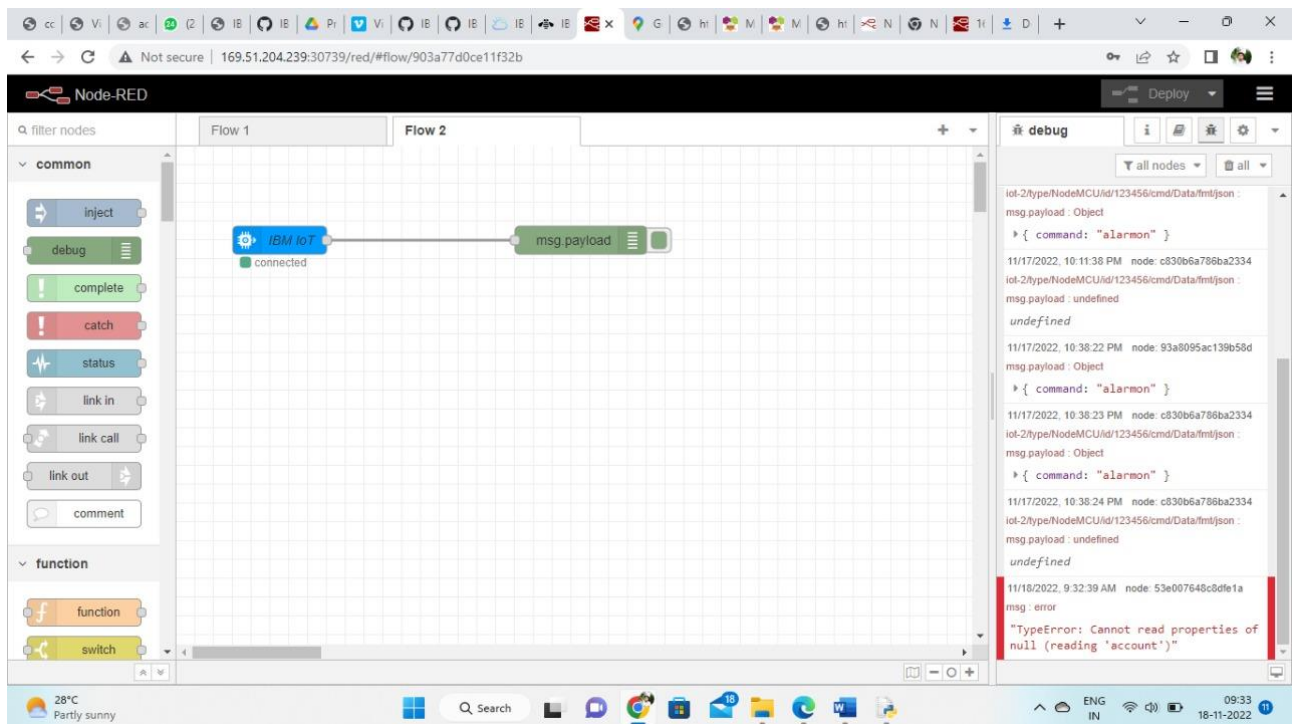
Authentication Method: use-token-auth

Authentication Token: gkfpv\_xfl1FB1)\*fvv

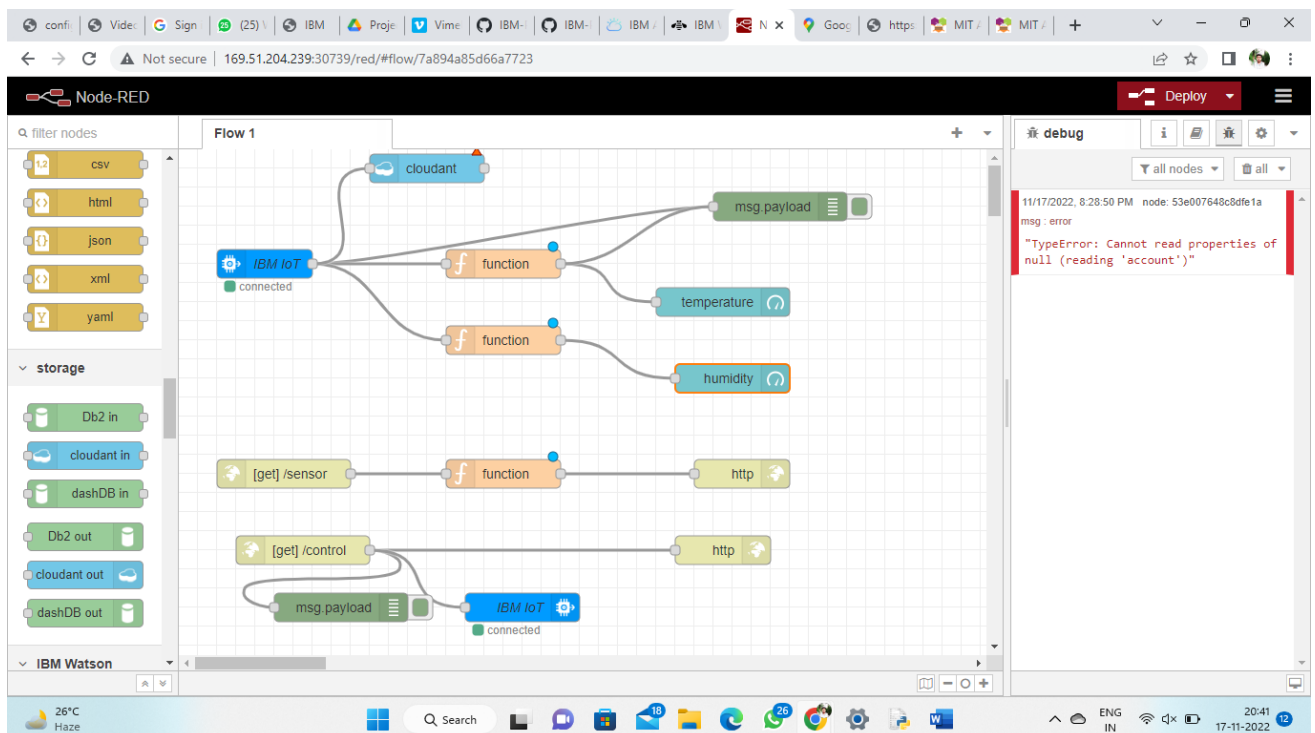
For initial testing, a node MCU device is created in IBM Watson IOT platform. Further the simulation is made run and Json data is seen in recent events of the same device.

The screenshot displays the IBM Cloud Developer console interface. The top navigation bar includes the IBM Cloud logo, a search bar, and links to Catalog, Manage, and the user profile (THIRUGNANA SAMPAN...). The main content area shows the details of a resource named 'Node RED ENRWX 2022-11-17'. The 'Details' section lists the App URL, Source, Resource group, Deployment target, and Created date. The 'Services' section shows the Cloudant service with links to Open dashboard, Documentation, and API reference. The 'Deployment Automation' section displays the deployment configuration, including Name, Location, Tool integrations, and Delivery Pipelines. The 'Delivery Pipelines' section shows two pipelines: 'ci-pipeline' with a 'Success' status and 'pr-pipeline' with a 'No stages detected' status. The bottom of the screen shows a Windows taskbar with various application icons and system information.

For the next process, the randomly generated values should be displayed in Node Red. So the Node red is configured.



## Configuring Node-RED:



- Once it is connected, Node-Red receives data from the device.
- Display the data using a debug node for verification.
- Connect the function node and write the Java script code to get each reading separately.

The Java script code for the function node is:

```
msg.payload=msg.payload.temp;
```

```
global.set('t', msg.payload)
```

```
return msg;
```

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

