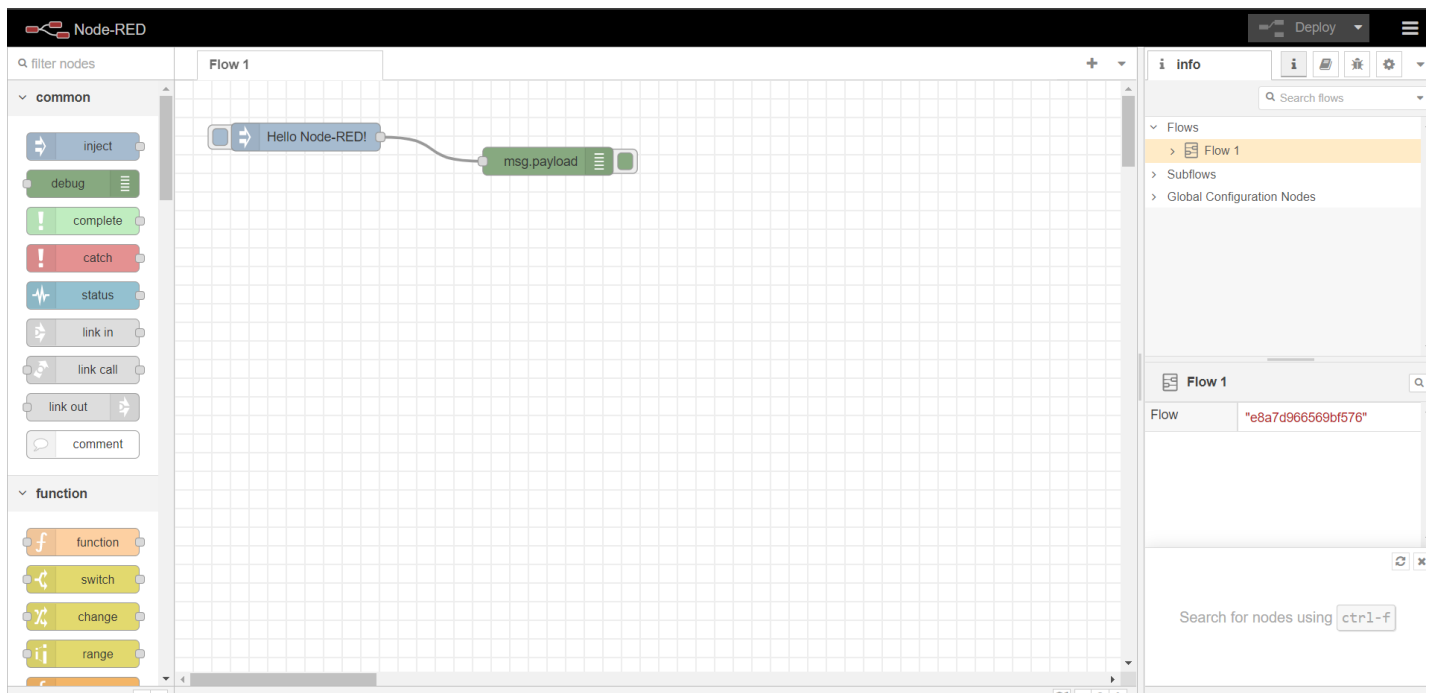


## Project Development phase SPRINT-2

|             |   |
|-------------|---|
| Date        | 02 NOVEMBER 2022                                |
| TEAMID      | PNT2022TMID46534                                |
| ProjectName | IOT Based Smart Crop Protection for Agriculture |
| Maximummark | 16 marks  |

### STEP1 : Setup Node-Red



## STEP 2 : Connect IBM IOT in and Debug1 and Deploy.

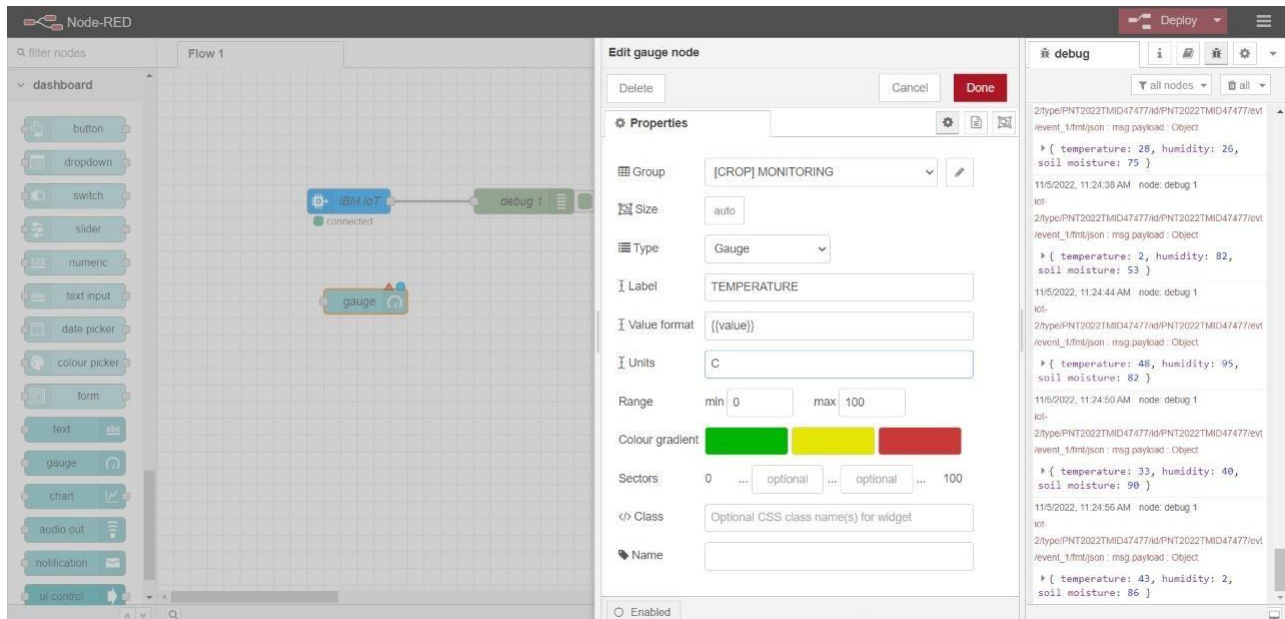
The image shows the Node-RED web interface. In the center workspace, a flow named 'Flow 1' contains two nodes connected in sequence: an 'IBM IoT' node (blue with a gear icon) and a 'debug 1' node (green with a list icon). The 'IBM IoT' node has a small green status indicator and the text 'connected' below it. On the left sidebar, the 'common' category is expanded, showing nodes like inject, debug, complete, catch, status, link in, link call, link out, and comment. The 'function' category is also expanded, showing function, switch, change, and range nodes. On the right sidebar, the 'debug' console is open, displaying a log of messages. The messages are JSON objects containing temperature, humidity, and soil moisture data, along with timestamps and node identifiers.

```
graph LR;
    subgraph Flow_1 [Flow 1]
        direction LR
        IoT[IBM IoT] --> Debug[debug 1]
    end
```

debug console output:

```
2/type/PNT2022TMID47477/ad/PNT2022TMID47477/evt
/evnt_1/fmt/json : msg.payload : Object
  { temperature: 86, humidity: 31,
    soil moisture: 54 }
11/5/2022, 11:20:36 AM node: debug 1
iot:
2/type/PNT2022TMID47477/ad/PNT2022TMID47477/evt
/evnt_1/fmt/json : msg.payload : Object
  { temperature: 8, humidity: 64,
    soil moisture: 59 }
11/5/2022, 11:20:39 AM node: debug 1
iot:
2/type/PNT2022TMID47477/ad/PNT2022TMID47477/evt
/evnt_1/fmt/json : msg.payload : Object
  { temperature: 98, humidity: 96,
    soil moisture: 53 }
11/5/2022, 11:20:44 AM node: debug 1
iot:
2/type/PNT2022TMID47477/ad/PNT2022TMID47477/evt
/evnt_1/fmt/json : msg.payload : Object
  { temperature: 96, humidity: 35,
    soil moisture: 25 }
11/5/2022, 11:20:50 AM node: debug 1
iot:
2/type/PNT2022TMID47477/ad/PNT2022TMID47477/evt
/evnt_1/fmt/json : msg.payload : Object
  { temperature: 78, humidity: 1,
    soil moisture: 28 }
```

**STEP 3 :** Edit gauge node (Here the gauge nodes are named as Temperature, Humidity and Soilmoisture).



## STEP 4 : PYTHON CODE

```
import time
import sys
import ibmiotf.application # to install pip install ibmiotf
import ibmiotf.device

#Provide your IBM Watson Device Credentials
organization = "hrodnmj" #replace the ORG ID
deviceType = "NODEMCU1" #replace the Device type wi
deviceId = "12345" #replace Device ID
authMethod = "token"
authToken = "kp1234" #Replace the authtoken

def myCommandCallback(cmd): # function for Callback
    print("Command received: %s" % cmd.data)
    if cmd.data['command'] == 'motoron':
        print("Motor On IS RECEIVED")

    elif cmd.data['command'] == 'motoroff':
        print("Motor Off IS RECEIVED")

    if cmd.command == "setInterval":

    if 'interval' not in cmd.data:
        print("Error - command is missing required information: 'interval'")
    else:
        interval = cmd.data['interval']
    elif cmd.command == "print":
```

```

if 'message' not in cmd.data:
    print("Error - command is missing required information: 'message'")
else:
    output=cmd.data['message']
    print(output)

try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method":
authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
    #.....

except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10
times
deviceCli.connect()

while True:

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud
deviceCli.disconnect()

```